Modeling for Control Design of an Axisymmetric Scramjet
Engine Isolator

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy in the Graduate School of The Ohio State
University

By

Alicia M. Zinnecker, B.S., M.S.

Graduate Program in
Electrical and Computer Engineering

The Ohio State University

2012

Dissertation Committee:

Prof. Andrea Serrani, Advisor
Dr. Jeffrey Donbar
Prof. Kevin Passino
Prof. Umit Ozguner
© Copyright by

Alicia M. Zinnecker

2012
Abstract

Renewed interest in scramjet propulsion has motivated efforts to construct models of the scramjet engine flow path that accurately capture transient flow dynamics relevant to unstart (in particular, the propagation of the shock wave due to a change in backpressure) and are amenable for use in designing a model-based controller that prevents unstart. Since unstart is an undesirable transient phenomenon, the engine model should capture transient flow dynamics to a desired level of accuracy while being amenable for control design. In this research, the modeling process is simplified by considering the engine as the cascade of an inlet/isolator subsystem and a combustor subsystem, with emphasis placed on constructing a model of the (axisymmetric) isolator. Four levels of model fidelity are considered in this effort: CFD models with high fidelity, which provide the data for constructing the less-accurate physics-based, locally-accurate, and data-based models. A series of physics-based isolator models is constructed through spatial discretization of the two-dimensional and one-dimensional Euler equations. The two models constructed from the two-dimensional Euler equations, a linearized model and a nonlinear model that has been modified to include suitable damping terms, are found to be too large to be simulated efficiently enough for use as a control-oriented model. The nonlinear model constructed through discretization of the one-dimensional Euler equations includes damping terms with an input-dependent scaling function. This model will be used in validation of the disturbance rejection controller designed for the data-based model constructed through
system identification with computational fluid dynamic solutions. Several linear isolator models, varying in order and input delay, are constructed and compared through simulation in order to evaluate the accuracy of each model in predicting the relationship between the shock location and the backpressure imposed on the isolator exit plane. The most accurate of these data-based models is chosen to form a candidate model set, for which the structured uncertainty in these models is transformed to unstructured uncertainty; the inability to accurately represent uncertainty in models of large order leads to the choice of a single data-based model for the control-oriented model. Two disturbance rejection controllers were designed for this model, using an $H_{\infty}$ control algorithm, with the goal of anchoring the shock wave at a specified location in the presence of input disturbances, one with stable dynamics and one with unstable dynamics. Both controllers were found to maintain the shock wave within 3 cm (26.1% of the duct height) of the specified location for input perturbations up to 13.789 kPa (2 psi). In analyzing the robust stability of the closed-loop model, it was found that the stable controller yielded a closed-loop model that was stable under larger model uncertainties than that constructed with the unstable controller. Validation of this controller in closed-loop with the physics-based model showed that it is able to maintain the shock location within 5 cm of the specified location for larger input perturbations tried; these simulations also revealed limitations in the accuracy of this model, including the inability to simulate the closed-loop system with a decrease in backpressure. The success of this controller in spite of these limitations shows the promise of applying this modeling-control design methodology to the problem of developing a controller to prevent unstart in a scramjet engine.
Acknowledgments

I would like to thank my adviser, Prof. Andrea Serrani, for all of the help and guidance as we worked through the many problems encountered, and for encouraging me when we hit roadblocks along the way, and Dr. Jeff Donbar at the Air Force Research Lab (AFRL) for all of his help in understanding the aerodynamics and flow dynamics that we were attempting to model and for providing insight from a different perspective in regards to this research. Thanks also to Dr. John Tam at the AFRL for agreeing to run the many sets of CFD data required to complete this research, the other engineers at the AFRL for offering their input when needed, and to the other students in the hypersonic vehicles research group for offering suggestions from outside eyes.

I would also like to thank the Dayton Area Graduate Studies Institute, The Ohio State University, and the Air Force Research Lab summer program for their support of my Master’s and Ph.D. research, including the purchase of a high-powered computer to complete the simulations required to validate the nonlinear model, and the Ohio Supercomputer Center for allowing me the use of their resources until this sufficiently-powerful computer was available.

Finally, I would like to thank my parents for the support and encouragement they’ve offered throughout the past five-and-a-half years. It has been a mix of ups and downs, and I appreciate the patience you showed with me, especially when I was
frustrated with the many seemingly-impassible obstacles encountered in my work and needed someone to vent to. It’s been a long road, but I’ve finally made it to the finish line.
Vita

1985 ........................................Born - Cincinnati, OH

2003-2007 ................................. B.S. Electrical Engineering  
                                      Valparaiso University  
                                      Valparaiso, IN

2007-2011 ................................. M.S. Electrical Engineering  
                                      The Ohio State University  
                                      Columbus, OH

2007-2008,2009-2011  ....................... DAGSI Fellowship

2008-2009,2011-2012  ....................... Distinguished University Fellowship

2011-2012 ................................. Ph.D. Candidate  
                                      Electrical Engineering  
                                      The Ohio State University  
                                      Columbus, OH

2012 ......................................... AFRL Fellowship

Fields of Study

Major Field: Electrical & Computer Engineering
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xvii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Modeling and control of scramjet engines: An overview of the state of the art</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Problem statement, methodology and contribution of the thesis</td>
<td>6</td>
</tr>
<tr>
<td>2. Physics-Based Model</td>
<td>11</td>
</tr>
<tr>
<td>2.1 CFD Grid and Data Collection</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Two-dimensional Euler equation-based model</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Continuity Equation</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2 Momentum Equations</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3 Energy Equation</td>
<td>18</td>
</tr>
<tr>
<td>2.2.4 Summary of Three-Dimensional Euler Equations</td>
<td>18</td>
</tr>
<tr>
<td>2.2.5 Derivation of an intermediate nonlinear model</td>
<td>19</td>
</tr>
<tr>
<td>2.2.6 Spatial discretization of the Euler equations</td>
<td>20</td>
</tr>
<tr>
<td>2.2.7 Constructing the Intermediate Nonlinear Model</td>
<td>28</td>
</tr>
<tr>
<td>2.2.8 Linearized model</td>
<td>39</td>
</tr>
<tr>
<td>2.2.9 Nonlinear model</td>
<td>69</td>
</tr>
<tr>
<td>2.3 One-dimensional Euler equation-based model</td>
<td>99</td>
</tr>
<tr>
<td>2.3.1 Damping modification</td>
<td>103</td>
</tr>
<tr>
<td>2.3.2 Model Validation</td>
<td>104</td>
</tr>
</tbody>
</table>
3. Data-based model

3.1 Modeling methodology

3.1.1 Data Collection and Preprocessing

3.1.2 Model Identification and Reduction

3.2 Model validation

3.2.1 Results

3.2.2 Constructing a candidate model set

3.3 Representing uncertainty in the model

3.3.1 Structured Uncertainty Representation

3.3.2 Unstructured Uncertainty Representation

3.3.3 Verifying Unstructured Representation

3.3.4 Refining the candidate model set

4. Control Design for Disturbance Rejection

4.1 Problem formulation

4.1.1 Open-loop model formulation

4.1.2 Closed-loop model formulation

4.1.3 Adding uncertainty to the model

4.1.4 Design goal

4.2 Control design and validation

4.2.1 Design and nominal stability

4.2.2 Robustness analysis

4.2.3 Validation on the reduced-order physics-based model

5. Conclusions and Outlook

5.1 Physics-based models

5.1.1 Outlook

5.2 Data-based model construction and control design

5.2.1 Outlook

References

Appendices

A. Two-dimensional Euler equation-based nonlinear model

A.1 Constructing the state matrix

A.2 Constructing the input matrix
B. One-dimensional Euler equation-based nonlinear model

B.1 Spatially discretizing the Euler equations

B.1.1 Modifications for boundary points

B.2 Constructing intermediate state equations

B.2.1 Building the intermediate state matrix

B.2.2 Building the nonlinear boundary condition input vector

B.3 Building the affine state equations

B.3.1 Constructing the state matrix

B.3.2 Constructing the input matrix

B.4 Constructing the nonlinear output function

B.5 Damping Modification

B.5.1 Constructing the Second-Order Damping Matrix

B.5.2 Building the Fourth-Order Damping Matrix
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Flowpath of a scramjet engine integrated with a hypersonic vehicle.</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Schematic of the four levels of model fidelity considered in problem.</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Block diagram of proposed cascade structure of the scramjet engine model.</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Three-dimensional grid of isolator for CFD simulation.</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Two-dimensional grid of isolator for modeling.</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Visualization of $A$ and $B$ matrices for the SPM.</td>
<td>53</td>
</tr>
<tr>
<td>2.4 Visualization of $C$ and $D$ matrices for the SPM.</td>
<td>53</td>
</tr>
<tr>
<td>2.5 Eigenvalues of $A$ for index scheme 1.</td>
<td>55</td>
</tr>
<tr>
<td>2.6 Schematics of mesh and index scheme from TecPlot360.</td>
<td>55</td>
</tr>
<tr>
<td>2.7 Comparison of index schemes 1 and 2 along the radial direction.</td>
<td>56</td>
</tr>
<tr>
<td>2.8 Eigenvalues of $A$ for index scheme 2.</td>
<td>57</td>
</tr>
<tr>
<td>2.9 Percent-errors of outputs at the center-line from simulation with $\delta u_1$.</td>
<td>60</td>
</tr>
<tr>
<td>2.10 Percent-errors of outputs at the radial midpoint from simulation with $\delta u_1$.</td>
<td>61</td>
</tr>
<tr>
<td>2.11 Percent-errors of outputs at the wall from simulations with $\delta u_1$.</td>
<td>61</td>
</tr>
<tr>
<td>2.12 Percent-errors of outputs at the center-line from simulation with $\delta u_2$.</td>
<td>63</td>
</tr>
</tbody>
</table>
2.13 Percent-errors of outputs at the radial midpoint from simulation with \( \delta u_2 \). ............................................... 63

2.14 Percent-errors of outputs at the wall from simulation with \( \delta u_2 \). .... 64

2.15 Radially-averaged perturbations in pressure, Mach number, and temperature profiles, from CFD data. .......................... 67

2.16 Pressure profiles from simulation of the undamped model. ........ 89

2.17 Flow variable profiles, at the center-line, from simulation of the undamped model. .................................................. 90

2.18 Flow variable profiles, at the wall, from simulation of the undamped model. .......................................................... 91

2.19 Pressure profiles from simulation of model with second-order damping. 92

2.20 Flow variable profiles, at the center-line, from simulation of the model with second-order damping. ......................... 93

2.21 Flow variable profiles, along the wall, from simulation of the model with second-order damping. .......................... 94

2.22 Pressure profiles from simulation of the model with second- and fourth-order damping. ................................. 95

2.23 Flow variable profiles, at the center-line, from simulation of the model with second- and fourth-order damping. ........ 96

2.24 Flow variable profiles, at the wall, from simulation of the model with second- and fourth-order damping. ................ 96

2.25 Pressure profiles from simulation of the model with second- and (modified) fourth-order damping. .......................... 97

2.26 Flow variable profiles, at the center-line, from simulation of the model with second- and (modified) fourth-order damping. ........ 97

2.27 Flow variable profiles, at the wall, from simulation of the model with second- and (modified) fourth-order damping. ........ 98
2.28 Pressure and flow variable profiles for initial conditions of simulations.

2.29 Expected final pressure and flow variable profiles for each input.

2.30 Pressure and flow variable profiles from simulation of the undamped model.

2.31 Pressure and flow variable profiles for simulation of the model with damping and an initial condition at tare.

2.32 Pressure and flow variable profiles from simulation of the model with $\alpha_{sc} = 500$.

2.33 Pressure and flow variable profiles from simulation of the model with $\alpha_{sc} = 400$.

2.34 Pressure and flow variable profiles from simulation of the model with $\alpha_{sc} = 400$ and $\beta_{sc} = 10000$.

2.35 Pressure and flow variable profiles from simulation of the model with $\alpha_{sc} = 400$ and $\beta_{sc} = 1000$.

2.36 Pressure and flow variable profiles from simulation of the model with initial condition $x_0^4$ and $\alpha_{sc} = 360$.

2.37 Pressure and flow variable profiles from simulation of the model with initial condition $x_0^4$, $\alpha_{sc} = 360$ and $p_{in} = p_{bD}$.

2.38 Relationship between expected steady-state shock location and $\alpha_{sc}$ for each input.

2.39 Accuracy of three fits for $\alpha_{sc}^+(p_{in})$.

2.40 Accuracy of three fits for $\alpha_{sc}^-(p_{in})$.

3.1 Block diagram of the scramjet engine, with a linear isolator model.

3.2 Input and output signals used for system identification and validation.

3.3 Pressure profiles and shock locations for experiment ‘stepA.’
3.4 Pressure profiles and shock locations for experiment ‘stepC.’ 144
3.5 ‘Best fit’ parameter for a large set of data-based models. 147
3.6 Calculation of AIC for various model orders and input delays. 149
3.7 Example Hankel singular value plot, for models in the ‘stepA’ family with \( nk = 1 \). 152
3.8 Hankel singular values for ‘stepB’ family models with \( nk = 2 \). 154
3.9 Eigenvalues of candidate models of the isolator. 155
3.10 Simulation results and output errors for validation against ‘stepA’ data. 159
3.11 Residual analysis for validation of the candidate models against ‘stepA’ data. 161
3.12 Simulation results and output errors for validation against ‘stepB’ data. 163
3.13 Residual analysis for validation of the candidate models against the ‘stepA’ data. 164
3.14 Simulation results and output errors for validation against ‘stepC’ data. 166
3.15 Residual analysis for validation of the candidate models against ‘stepC’ data. 167
3.16 Variation of elements in \( A \) due to model uncertainty. 179
3.17 Variation of elements in \( B \) due to model uncertainty. 180
3.18 Variation of elements in \( C \) due to model uncertainty. 180
3.19 Magnitude responses of \( (W \Delta)_{\text{mult}} \) for the ‘stepA’ family model with \( n = 5, nk = 1 \). 182
3.20 Magnitude responses of \( (W \Delta)_{\text{mult}} \) for the ‘stepA’ family model with \( n = 30, nk = 1 \). 183
3.21 Magnitude responses of \( (W \Delta)_{\text{mult}} \) for the ‘stepB’ family model with \( n = 30, nk = 1 \). 184

xiii
3.22 Magnitude responses of \((W\Delta)_{\text{mult}}\) for the ‘stepB’ family model with 
\(n = 25, nk = 2\). .......................................................... 184

3.23 Frequency responses of uncertain model formulations of the \(n = 5, nk = 1\) model in the ‘stepA’ family. ................................. 186

3.24 Frequency responses of uncertain model formulations of the \(n = 30, nk = 1\) model in the ‘stepA’ family. ................................. 187

3.25 Frequency responses of uncertain model formulations of the \(n = 30, nk = 1\) model in the ‘stepB’ family. ................................. 188

3.26 Frequency responses of uncertain model formulations of the \(n = 25, nk = 2\) model in the ‘stepB’ family. ................................. 188

3.27 Frequency responses of models with uncertainty determined through method 2. ............................................................ 190

3.28 Magnitude responses used to construct \(W_1\) using \(ucover\) for the \(n = 5, nk = 1\) ‘stepA’ family model. ................................. 191

3.29 Magnitude responses used to construct \(W_1\) using \(ucover\) for the \(n = 30, nk = 1\) ‘stepA’ family model. ................................. 192

3.30 Magnitude responses used to construct \(W_1\) using \(ucover\) for the \(n = 30, nk = 1\) ‘stepB’ family model. ................................. 192

3.31 Magnitude responses used to construct \(W_1\) using \(ucover\) for the \(n = 25, nk = 2\) ‘stepB’ family model. ................................. 193

3.32 Comparison of uncertain \(n = 5, nk = 1\) models in the ‘stepA’ family. 195

3.33 Comparison of uncertain \(n = 30, nk = 1\) models in the ‘stepA’ family. 197

3.34 Comparison of uncertain \(n = 30, nk = 1\) models in the ‘stepB’ family. 198

3.35 Comparison of uncertain \(n = 25, nk = 2\) models in the ‘stepB’ family. 198

3.36 Comparison of \(|W|\) constructed for each discrete-time model and each methodology. ......................................................... 200
3.37 Comparison of $|W|$ constructed for each continuous-time model and each methodology. .................................................. 201
3.38 Frequency responses for each uncertain discrete-time model. ................. 203
3.39 Frequency responses for each uncertain continuous-time model. ............... 203
4.1 Block diagram of the (nominal) closed-loop DBM. ................................. 208
4.2 Block diagram of the closed-loop DBM with multiplicative uncertainty. .... 212
4.3 Shock wave perturbations for simulation of closed-loop nominal model. .... 219
4.4 Shock wave locations for simulations of uncontrolled and controlled nominal model. ............................................................... 220
4.5 Input disturbance and control effort for simulation of closed-loop nominal model. ................................................................. 222
4.6 Bode plots of modified uncertain (open-loop) model. ............................... 223
4.7 Shock perturbations for simulation of the closed-loop uncertain model. .... 224
4.8 Transient behavior of shock perturbations for simulation of the closed-loop uncertain model. ....................................................... 225
4.9 Input disturbances and control efforts for simulation of the closed-loop uncertain model. ......................................................... 226
4.10 Transient behavior of input disturbances and control efforts for simulation of the uncertain model with unstable controller. ............... 227
4.11 Transient behavior of input disturbances and control efforts for simulation of the uncertain model with stable controller. ............... 227
4.12 Block diagram of the closed-loop PBM. ............................................ 228
4.13 Shock locations for simulation of the closed-loop PBM with the stable disturbance rejection controller. ................................................... 230
4.14 Control efforts for simulation of the closed-loop PBM with the stable disturbance rejection controller. ............................................. 232
4.15 Shock location and control effort for simulation of the closed-loop PBM with stable disturbance rejection controller and modified initial conditions.
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Governing equations applied to each point in the two-dimensional CFD grid.</td>
</tr>
<tr>
<td>2.2</td>
<td>Run times for each SPM simulation.</td>
</tr>
<tr>
<td>2.3</td>
<td>Streamwise locations of sensors for SPM.</td>
</tr>
<tr>
<td>2.4</td>
<td>Steady-state percent-errors of SPM outputs from simulations with $\delta u_1$ and $\delta u_2$.</td>
</tr>
<tr>
<td>2.5</td>
<td>Scaling factors necessary to achieve accurate steady-state results.</td>
</tr>
<tr>
<td>2.6</td>
<td>Backpressure and shock-location for each steady-state condition available from CFD.</td>
</tr>
<tr>
<td>2.7</td>
<td>Scaling factors necessary to achieve accurate steady-state shock locations for each $(x_0, p_{in})$ pair.</td>
</tr>
<tr>
<td>2.8</td>
<td>Relative error in shock location for tuned scaling factors, for each $(x_0, p_{in})$ pair.</td>
</tr>
<tr>
<td>2.9</td>
<td>Coefficients of fits for $\alpha_{sc}(p_{in})$.</td>
</tr>
<tr>
<td>2.10</td>
<td>$\alpha_{sc}$ estimated by each fit, for each input.</td>
</tr>
<tr>
<td>2.11</td>
<td>Relative shock location error for simulation with each $(x_0, p_{in})$ pair and fit 1.</td>
</tr>
<tr>
<td>2.12</td>
<td>Relative error in $\alpha_{sc}$ for simulation with each $(x_0, p_{in})$ pair and fit 1.</td>
</tr>
</tbody>
</table>
# NOMENCLATURE

## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4SID</td>
<td>state-space subspace identification technique</td>
</tr>
<tr>
<td>AFRL</td>
<td>Air Force Research Lab</td>
</tr>
<tr>
<td>AIC</td>
<td>Akaike information criterion</td>
</tr>
<tr>
<td>CI</td>
<td>confidence interval</td>
</tr>
<tr>
<td>CFD</td>
<td>computational fluid dynamics</td>
</tr>
<tr>
<td>COM</td>
<td>control-oriented model</td>
</tr>
<tr>
<td>CT</td>
<td>continuous time domain</td>
</tr>
<tr>
<td>DBM</td>
<td>data-based model</td>
</tr>
<tr>
<td>DT</td>
<td>discrete time domain</td>
</tr>
<tr>
<td>FSM</td>
<td>full-scale model</td>
</tr>
<tr>
<td>H.O.T.</td>
<td>higher-order terms (in approximations)</td>
</tr>
<tr>
<td>HSV</td>
<td>Hankel singular values</td>
</tr>
<tr>
<td>LTI</td>
<td>linear time invariant model</td>
</tr>
<tr>
<td>MIMO</td>
<td>multi-input, multi-output model</td>
</tr>
<tr>
<td>N4SID</td>
<td>numerical state-space subspace identification technique</td>
</tr>
<tr>
<td>PBM</td>
<td>physics-based model</td>
</tr>
<tr>
<td>PCT</td>
<td>Parallel Computing Toolbox (MATLAB®)</td>
</tr>
<tr>
<td>POD</td>
<td>proper orthogonal decomposition</td>
</tr>
<tr>
<td>RCT</td>
<td>Robust Control Toolbox (MATLAB®)</td>
</tr>
<tr>
<td>ROM</td>
<td>reduced-order model</td>
</tr>
<tr>
<td>SIDT</td>
<td>System Identification Toolbox (MATLAB®)</td>
</tr>
<tr>
<td>SISO</td>
<td>single-input, single-output model</td>
</tr>
<tr>
<td>SPM</td>
<td>small-perturbation model</td>
</tr>
<tr>
<td>SVD</td>
<td>singular value decomposition</td>
</tr>
</tbody>
</table>
**Variables (uppercase)**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A(x)$</td>
<td>state matrix in nonlinear affine and linear models; dependence on $x$ is dropped for linear models</td>
</tr>
<tr>
<td>$\bar{A}(x)$</td>
<td>intermediate state matrix in nonlinear model</td>
</tr>
<tr>
<td>$A(\cdot)$</td>
<td>submatrix of $A(x)$ for grid point given by subscripted index; velocity dependence may also be specified</td>
</tr>
<tr>
<td>$A^*(x)$</td>
<td>state matrix in realization of four-block problem ($H_\infty$ control)</td>
</tr>
<tr>
<td>$B(x)$</td>
<td>input matrix in nonlinear affine and linear models; dependence on $x$ is dropped for linear models</td>
</tr>
<tr>
<td>$\bar{B}(x,p_{in})$</td>
<td>intermediate input matrix in nonlinear model</td>
</tr>
<tr>
<td>$B(\cdot)$</td>
<td>submatrix of $B(x)$ for grid point given by subscripted index; velocity dependence may also be specified</td>
</tr>
<tr>
<td>$B(\cdot)_{bc,(\cdot)}$</td>
<td>input matrix partition in realization of four-block problem ($H_\infty$ control)</td>
</tr>
<tr>
<td>$C(x)$</td>
<td>output matrix in nonlinear affine and linear models dependence on $x$ is dropped for linear models</td>
</tr>
<tr>
<td>$C(\cdot)$</td>
<td>submatrix of $C(x)$ for grid point given by subscripted index</td>
</tr>
<tr>
<td>$C(\cdot)$</td>
<td>output matrix partition in realization of four-block problem ($H_\infty$ control)</td>
</tr>
<tr>
<td>$C$</td>
<td>vector of unknown curve-fit coefficients</td>
</tr>
<tr>
<td>$D$</td>
<td>feedthrough matrix in nonlinear affine and linear models</td>
</tr>
<tr>
<td>$D(\cdot)$</td>
<td>submatrix of $D$ for grid point given by subscripted index</td>
</tr>
<tr>
<td>$D_H$</td>
<td>duct height, $D_H = 2R(z)$ (average value 11.5 cm)</td>
</tr>
<tr>
<td>$D^2_r, D^4_z$</td>
<td>second- or fourth-order damping functions in radial ($r$) and streamwise ($z$) directions</td>
</tr>
<tr>
<td>$E$</td>
<td>specific total energy (flow variable)</td>
</tr>
<tr>
<td>$G$</td>
<td>transfer function (or realization) for the four-block problem defined for $H_\infty$ control</td>
</tr>
</tbody>
</table>
\( J \) = Hamiltonian matrix representing ‘observability’ Riccati equation 
\((H_\infty \text{ control})\)

\( J_{(\cdot)} \) = Jacobian of the flux function indicated in the subscript

\( J_{F(\cdot)}^z \) = flux-split Jacobian calculated from \( J_{(\cdot)} \)

\( J_{F(\cdot)}^{\pm} \) = submatrix of \( A(x) \) containing the Jacobians \( J_{F}^{\pm} \) for grid point 
given by subscripted index

\( J_{\text{mod}} \) = matrix modifying \( A(x) \) from boundary condition treatment 
(subsonic exit flow)

\( H \) = enthalpy (flow variable)

\( H \) = Hamiltonian matrix representing ‘controllability’ Riccati equation 
\((H_\infty \text{ control})\)

\( K \) = transfer function (or realization) for the controller

\( K_{(\cdot)} \) = eigenvector matrix of the Jacobian of the flux function in 
subscript

\( K_{(\cdot)}^{-1} \) = inverse eigenvector matrix of the Jacobian of the flux function 
in subscript

\( L \) = matrix modifying Jacobians in boundary condition treatment 
(subsonic exit flow)

\( L \) = isolator length (90.678 cm)

\( L_B \) = matrix containing input terms in boundary condition treatment 
(subsonic exit flow)

\( N_r, N_z \) = number of points in the isolator grid along the radial \((r)\) and 
streamwise \((z)\) axes

\( P \) = transfer function of the model (plant); presence of uncertainty 
indicated by subscript \( \Delta \)

\( P^2(x) \) = second-order damping matrix in nonlinear affine model

\( P^2_{(\cdot)}(x) \) = submatrix of \( P^2(x) \) for grid point given by subscripted index; 
location in \( P^2(x) \) is also specified

\( P^4(x) \) = fourth-order damping matrix in nonlinear affine model

\( P^4_{(\cdot)}(x) \) = submatrix of \( P^4(x) \) for grid point given by subscripted index; 
location in \( P^4(x) \) is also specified

\( P_{in}, P_{in,i} \) = (sub)matrix of powers of \( p_{in} \) (for curve-fitting)

\( R \) = universal gas constant

\( R(z) \) = radius of the isolator
\( T \) = temperature (flow variable)  
\( T_{x,y} \) = transfer function from input \( x \) to output \( y \)  
\( V \) = loss function for identification of a model with a specific structure  
\( W \) = weighting matrix for uncertain dynamics in a model with uncertainty  
\( W_{1,2,3} \) = shaping filters augmented for definition of four-block problem (\( H_\infty \) control)  
\( X_\infty, Y_\infty \) = solutions of the Riccati equations defined by \( H \) and \( J \) (\( H_\infty \) control)  

**Variables (lower case)**  
\( c \) = sonic velocity (flow variable)  
\( f(u) \) = flux function associated with partial derivative w.r.t. \( r \)  
\( h(u) \) = flux function associated with partial derivative w.r.t. \( z \)  
\( (i, k) \) = index of a point in the isolator grid; \( i \) indexes \( r \)-axis, \( k \) indexes \( z \)-axis  
\( j(u) \) = flux function containing terms not associated with partial derivatives (two-dimensional Euler equations)  
\( k(u) \) = flux function associated with cross-sectional area-dependent terms (one-dimensional Euler equations)  
\( k(x) \) = nonlinear function associated with non-feedthrough terms in output  
\( k_{(i)}(u) \) = component of \( k(x) \) associated with grid point indexed in subscript; primed function indicates exit plane  
\( l(x) \) = nonlinear function associated with feedthrough terms in output  
\( l_{(i)}(u) \) = component of \( l(x) \) associated with grid point indexed in subscript  
\( m \) = input dimension of model  
\( n \) = state dimension of model  
\( nk \) = input delay/relative degree of model  
\( p \) = pressure (flow variable)  
\( p_b, \tilde{p}_b \) = backpressure, backpressure disturbance  
\( p_{in} \) = input to nonlinear affine and linearized model (backpressure)  
\( q \) = output dimension of model
(r, θ, z) = cylindrical coordinate triplet
$u_{(.)}$ = state variables associated with subscripted index; subscript may be dropped
$u$ = control input
$u_{(.)}$ = component of $u$
$u_p$ = input to plant
$u_r$ = radial component of flow velocity (flow variable)
v = flow velocity vector (flow variable)
v$_\theta$ = azimuthal component of flow velocity (flow variable)
w = streamwise component of flow velocity (flow variable)
x = state vector of nonlinear affine and linear models
$x_{(.)}^0$ = initial condition for simulation
$y$ = output vector of nonlinear affine and linear models
$z_{1,2,3}$ = outputs measured from each shaping filter in four-block problem
$z_{sh}$, $\tilde{z}_{sh}$ = shock location (shock perturbation)

Variables (Greek)

$A_{sc,i}$, $A_{sc}$ = (sub)matrix of scaling factors (curve-fitting)
$\Delta$ = system with uncertain dynamics
$\Delta^{(.)}$ = represents the forward-difference ($f$) or backwards-difference ($b$) operator
$\Delta T$ = time-step used in CFD simulation
$\Lambda_{\pm}^{(.)}$ = diagonal matrix containing positive or negative eigenvalues of the Jacobian of the flux function indicated in the subscript
$\alpha_{(.)}^{(*)}$ = product of $\lambda$ and $\epsilon^2$ for the given index shift; * indicates boundary condition modification
$\alpha_{(.)}^{(.)}$ = sum of the coefficients on $u_{i\pm 1,k}$ and $u_{i,k\pm 1}$ in the second-order damping terms
$\alpha_{sc}$ = scaling factor on second-order damping matrix (fixed)
$\alpha_{sc}^{(p_{in})}$ = input-dependent scaling factor on second-order damping matrix
$\beta_{(.)}^{(*)}$ = product of $\lambda$ and $\epsilon^4$ for the given index shift;
* indicates boundary condition modification
\[ \beta^{(\cdot)} = \text{sum of the coefficients on } u_{i\pm2,k}, u_{i\pm1,k}, u_{i,k\pm2}, \text{ and } u_{i,k\pm2} \text{ in the fourth-order damping terms} \]

\[ \beta_{sc} = \text{scaling factor on fourth-order damping matrix (fixed)} \]

\[ \gamma = \text{specific heat ratio} \]

\[ \delta(\cdot) = \text{perturbation of the flow variable} \]

\[ \epsilon = \text{feedthrough scaling for filter } W_2 \text{ (control input) in four-block problem (} H_\infty \text{ control)} \]

\[ \epsilon^{2,4} = \text{scaling factor in damping terms related to the pressure gradient} \]

\[ \epsilon(\cdot) = \text{prediction error calculated from simulation output} \]

\[ \eta = \text{feedthrough scaling for filter } W_3 \text{ (shock perturbation) in four-block problem (} H_\infty \text{ control)} \]

\[ \kappa^{2,4} = \text{scaling factors for second- and fourth-order damping terms} \]

\[ \lambda(\cdot) = \text{scaling factor in damping terms related to the spectral radius of Jacobians} \]

\[ \nu(\cdot) = \text{pressure gradient-dependent term in damping modification} \]

\[ \xi = \text{feedthrough scaling for filter } W_1 \text{ (additive sensor noise) in four-block problem (} H_\infty \text{ control)} \]

\[ \rho = \text{density (flow variable)} \]

\[ \sigma(\cdot) = \text{uncertainty related to the standard deviation of matrix elements or polynomial coefficients} \]

**Sub/Superscripts**

\[ \text{CL} = \text{system matrix or parameter for closed-loop model realization} \]

\[ \text{G, G(\cdot)} = \text{system matrix or partition for four-block problem realization (} H_\infty \text{ control)} \]

\[ \text{add, A} = \text{weighting matrix, plant associated with additive uncertainty representation} \]

\[ \text{d} = \text{submatrices located along the block diagonal of a larger matrix} \]

\[ \text{fit} = \text{weighting matrix resulting from a fit to data} \]

\[ \text{k} = \text{system matrix or parameter for controller realization} \]

\[ \text{mult, M} = \text{weighting matrix, plant associated with multiplicative uncertainty representation} \]

\[ \text{offset} = \text{represents an offset from steady-state} \]

xxiv
$\text{sub} = \text{matrix (submatrix) or function associated with subsonic flow}$

$= \text{submatrices located along the block superdiagonal of a larger matrix}$

$\text{sup} = \text{matrix (submatrix) or function associated with supersonic flow}$

$= \text{submatrices located along the block superdiagonal of a larger matrix}$

$\Delta = \text{system contains uncertainty}$

$\sigma = \text{matrix contains elements with uncertainty related to the standard deviation}$

$\ast = \text{matrix or state associated with a ‘set-point’}$

$= \text{index shift for a state or flow variable}$

$\bar{\cdot} = \text{steady-state value of the flow variable}$

$\hat{\cdot} = \text{effective state matrix when all modifications are considered}$

$\hat{\cdot} = \text{expected output calculated from CFD data}$
Chapter 1: Introduction

For nearly a century, the aerospace community has been seeking to achieve successful hypersonic flight. Rene Lorin presented the first design of a ram compression air-breathing engine (for subsonic flight) in 1913 and in 1928 Albert Fono’s conceptual ramjet engine for supersonic flight followed. Rene Luduc’s ramjet design was patented in 1935 and successfully flown in 1949; the vehicle reached a speed of Mach 0.9 in subsequent flights, ushering in two decades of research, design, and testing of turbo-ramjet, ramjet and scramjet engines [1].

The high point of early scramjet research occurred in the 1950’s as scientists and engineers sought to improve the performance of ramjets when flown at hypersonic speeds by modifying the engine design. NASA experiments and engine component design studies lead to the construction of the hypersonic research engine (HRE), which was incorporated into the X-15 project. Curran and Stull introduced the idea of a dual-mode ramjet-scramjet engine in 1963 and received a patent for the design in 1969 [2]; results of the engine operating in ramjet mode were published by Billig in 1966 [3, 4]. Unfortunately, the X-15 project was canceled in 1968 due to costs associated with the project, and interest in air-breathing propulsion lessened by the 1970’s, perhaps due in part to project costs, but because the necessary technology had not been developed yet [1].

Interest in high-speed air-breathing engines, especially scramjets, has recently picked up. Hypersonic air-breathing propulsion was brought back to the forefront by
the German Sänger Space Transportation system and NASP aerospace plane projects, where it has since remained [1]. The recent successful flights of NASA’s X-43 vehicle and the X-51 Waverider have marked milestones in the development of scramjet technology [5].

High-speed air-breathing engines, such as ramjets and scramjets, offer advantages over the more commonly-used rocket engines. Air-breathing engines do not have the rotating machinery found in rocket engines, and they do not require that an oxidizer be carried on board with the fuel. This allows for an increase in payload or smaller vehicle airframe, the latter of which would decrease the design, building and launch costs of hypersonic vehicles, since oxidizers constitute a large percentage of the payload in traditional launch vehicles [1]. Ramjet and scramjet engines are designed to operate at supersonic flight speeds, producing thrust from the ram pressure created by decelerating the airflow. Air is decelerated to subsonic speeds for combustion in ramjet engines, a process that is inefficient and leads to a decrease of thrust at hypersonic flight speeds (around Mach 6). This performance loss is recovered in scramjet engines, as the air remains supersonic throughout the engine flow path, including the combustor [1].

The scramjet engine is often integrated with the vehicle air frame, which allows for the fore- and aft-bodies of the vehicle to be utilized as external compression and expansion surfaces to decelerate and accelerate the air as it enters and exits the engine. A representative scramjet flowpath for this integrated geometry is shown in Figure 1.1. The air is compressed externally as it passes through the inlet (1-2), then internally compressed in the isolator (2-3) before the flow enters the combustor. The air flow decelerates, and the pressure of the air entering the combustor increases. When the combustor is ignited, a normal or oblique shock train forms in the isolator, depending on the freestream velocity; the oblique shock train shown in Figure 1.1
is typical for a scramjet engine flow path. As heat is added in the combustor, the flow is decelerated and the pressure increases. In the nozzle, the flow expands and accelerates, producing thrust to propel the vehicle.

Unfortunately, scramjet engines are characterized by complex flow and combustion phenomena, which are not well understood from a physics-standpoint and, therefore, are difficult to model. The possibility of unstart, specifically in low-velocity parts of the flight envelope (flight Mach number less than 5), presents another hurdle in improving the operability of these air-breathing engines as it leads to undesirable operating conditions. A scramjet engine can become unstarted when the shock train that has formed in the isolator of a started engine becomes dislodged and reforms external to the engine inlet. Typically, this is caused by perturbations in the flow related to an increase in engine backpressure during acceleration or flight maneuvers. In addition to a decrease in thrust production, lower total pressure in the flow, and problems in the combustor [1], unstart also produces unbalanced pitching moments that can lead to instability in an integrated vehicle-airframe configuration. At the onset of unstart, the instantaneous loss of thrust is usually accompanied by sudden changes in pitching moment, typically due to slow reversal in stability and control
derivatives, and residual side forces and yawing/rolling moments [6]. Unless proper corrective actions are taken, inlet unstart may result in unrecoverable system failure. The characterization and incorporation of limits of operability of the scramjet engine in the flight control system was considered in Refs. [7, 8] for the vehicle model of Bolender and Doman [9]. In the cited references, the development of appropriate control actions for the generation and tracking of reference trajectories that avoid the occurrence of unstart was also addressed. However, preemptive or corrective actions taken by the vehicle guidance and control system alone may not suffice to prevent the occurrence of engine unstart. Active control of the engine becomes necessary to prevent or reverse unstart, while optimizing engine performance during acceleration or high-demand maneuvers. It is also possible that being able to design a controller at the engine level which tightly controls the position of the shock train in the isolator will reduce the necessary length of the isolator, and perhaps minimize the need for this component, which would decrease the drag of the engine.

1.1 Modeling and control of scramjet engines: An overview of the state of the art

In order to construct a model-based controller, it is necessary to first have a small order model that accurately captures the complex flow dynamics of the engine. Due to the complexity of scramjet dynamics, several assumptions and simplifications must be made during the modeling process. Early models, like that derived by Culick and Rogers [10], considered only pre-combustion sections of the engine (that is, the inlet) operating in ramjet mode, since little data was available about the flow phenomena associated with oblique shock trains and combustion chemistry. A model-based feedback controller, designed by MacMartin, was shown, through simulation of the Culick
and Rogers model, to anchor the shock wave in the presence of flow perturbations that caused unstart to occur in uncontrolled flow [11].

As experimental observation and computational fluid dynamics (CFD) simulation improved the understanding of the shock train-related flow phenomena, other models have emerged which have limited application to control design. Experimental studies by Lin, et al. led to the derivation of correlations between Mach number and pressure ratios for two different isolator geometries, and a single shock profile description [12] while qualitative unstart ‘margins’ for velocity, pressure, and temperature perturbations were developed by Mayer and Paynter [13]. Both models describe steady-state flow conditions and do not capture the transition dynamics that are important in designing a controller. A recent effort at the Michigan/Air Force Collaborative Center for Control Science (MACCS) concentrated on developing a steady-state model of a scramjet engine, called MASIV, to replace the simple engine model implemented in the Bolender and Doman hypersonic vehicle model [9]; in this model, tracking and guidance controllers have been designed that use the engine as an actuator [7, 8, 14]. Although the MASIV model includes a sophisticated subsystem for modeling and designing the inlet [15] and a complex combustor model that contains mixing chemistry, injection jet dynamics, and simple chemical kinetics [16–21], it does not capture the transient dynamics of the flow and therefore cannot be used to design a controller for preventing unstart, which may occur during transitions between steady-states. In addition, the engine should be considered a subsystem of the vehicle, rather than as a control input to the overall system, so a model is necessary that captures the engine dynamics due to actuation during operation.

Recent collaboration between the Air Force Research Lab (AFRL) and Air Force Institute of Technology (AFIT) has focused on development of a methodology for estimating the location of the shock wave leading-edge and identification of a dynamic
model that captures shock propagation due to changing backpressure using data collected from wind tunnel experiments of a cold-flow isolator [22, 23]. These dynamic models were constructed for an isolator with rectangular cross section and shown to be accurate in predicting the shock location when validated in a real-time experimental set-up, suggesting they could possibly act as control-oriented model. In collaboration with the AFRL, a group at the University of Texas at Austin has been performing similar investigations involving the detection and data-based modeling of unstart in a cold-flow isolator [24–27]. The data-based modeling effort discussed in Ref. [27] use pressure data collected from four different experimental runs, at two different facilities, to construct a series of models shown to predict the pressure measurements for a given backpressure function. An alternative to these data-based algorithms is to consider a physics-based model (PBM) such as that proposed by Chicatelli and Hartley [28, 29], where the isolator model is constructed from the Euler equations. A similar formulation is presented by Bassi and Rebay in Ref. [30] and extended to the more complex Navier-Stokes equations in Ref. [31]. This methodology involves discretization of the governing equations to construct a nonlinear model in affine form; linearization of these equations then yields a model related to that constructed by Chicatelli and Hartley [28, 29]. A similar methodology, where a reduced-order model is constructed from the Navier-Stokes equations using a finite-element method is developed by Ito and Ravindran [32], where steady-state and time-dependent models are constructed and validated with classical test cases.

1.2 Problem statement, methodology and contribution of the thesis

The ultimate goal of this thesis is the designing of a model-based controller that will act to prevent, or reverse, the occurrence of unstart in scramjet engines. With
the lack of literature available addressing modeling of a scramjet engine, this effort focuses on building a model that captures the dynamics of flow through the engine to a level of accuracy that is suitable for control design, while being computationally efficient to allow model-based control. In this control-oriented modeling process, four levels of model fidelity are considered, as depicted in Figure 1.2. The high-fidelity models - CFD and, if available, experimental data - provide final validation of the modeling and control design methodologies. The three lower-fidelity models capture relevant flow field dynamics with less accuracy than the high-fidelity models while lending themselves to more efficient simulation (physics-based and locally-accurate model) and control design (control-oriented model).
Figure 1.3: Block diagram depicting the proposed cascade structure of the scramjet engine model. The combustor subsystem models the relationship between an actuator and backpressure (the input to the isolator subsystem). For the physics-based model, the location of the shock wave, \( y = z_{sh} \), is estimated from a pressure profile.

For simplicity, the engine model will be considered in the cascade form shown in Figure 1.3, where the isolator subsystem models the relationship between back-pressure and shock location and the combustor subsystem models the relationship between an actuator and the backpressure imposed on the isolator. This modeling effort focuses on the isolator, where two approaches will be considered: a physics-based modeling (PBM) approach, where an isolator model is constructed through spatial discretization of the compressible Euler equations, and a data-based model (DBM) developed using system identification techniques and transient CFD data.

The PBM approach described in Chapter 2 considers first a linearized model of the isolator constructed from the two-dimensional Euler equations; this model was shown in simulation to be unable to capture the flow dynamics, so a nonlinear PBM was considered. The size of the nonlinear model resulting from discretization of the two-dimensional Euler equations was found to be prohibitive for simulation, so the one-dimensional Euler equations were used to construct a model that is useful for control design validation. The DBM described in Chapter 3 is a small-order linear model constructed from transient CFD data to capture the motion of the leading edge.
of the shock wave in response to step changes in the backpressure imposed on the isolator exit. Since the CFD data used to construct the DBM is from inviscid flow simulations, a controller designed for this model can be implemented with the PBM, while the small size of the DBM model allows for quicker simulation (for initial control design validation) than the PBM. Uncertainty in the model parameters, due to high-frequency dynamics that are not modeled during system identification, can be used to develop a linear model with multiplicative uncertainty to which robust analysis techniques can be applied. In Chapter 4, two disturbance rejection controllers are designed using an $H_\infty$ control design technique to minimize the influence of an input disturbance on the output of the DBM; one controller is stable, the other is unstable. Both controllers are validated on the DBM, and it is shown that the stable controller is robust against any perturbation in the model within the range of uncertainty; the unstable controller produces a closed-loop model that is robust against a smaller range of perturbation. The stable controller is further validated through implementation with the PBM, where it is shown to perform as expected despite limitations in the accuracy of the model.

The results in this thesis represent a first attempt in construction of a control-oriented model of a scramjet engine isolator using several available tools: CFD simulation, physics-based modeling methodologies, and data-based modeling methodologies. Although the usefulness of the physics-based model is limited in accuracy, the modifications made to the model, specifically the introduction of damping matrices (with an input-dependent scaling function) to the nonlinear model, and the modeling methodology, have improved the linearized model introduced by Chicatelli and Hartley in Ref. [28] and provide an important first-step toward construction of an accurate, control-oriented physics-based model. The data-based modeling effort uses transient CFD data, instead of input-output data collected from experimental test
runs of an engine flow path, yields an accurate, small-order linear model that cap-
tures shock propagation due to changes in backpressure. The size and structure of
this model lends itself to control methodologies that take uncertainty into account
directly, such as the $H_\infty$ design methodology used to construct a disturbance rejec-
tion controller; even when uncertainty was taken into consideration, the closed-loop
model was of a small enough size to be computationally inexpensive. The controller
was found to successfully meet the design goal when implemented in closed-loop with
the data-based model (for which it was designed) and in closed-loop with the physics-
Based model, despite the limitations of this higher-fidelity model, which shows the
possibility of a model-based controller being effective in preventing unstart when im-
plemented in a model of higher fidelity than that for which it was designed. These
modeling and control efforts provide a first-step toward the design of a model-based
controller that can then be validated through implementation with a higher-fidelity
model and represent the first model-based feedback controller designed to meet the
goal of preventing unstart in a scramjet engine.
Chapter 2: Physics-Based Model

A physics-based model (PBM) of the isolator is constructed through spatial discretization of the compressible Euler equations following the modeling technique developed by Chicatelli and Hartley [28, 29]. Bassi and Rebay present a similar formulation in Ref. 30 and extend this technique to the more complex Navier-Stokes equations in Ref. 31, but in the interest of simplicity the inviscid Euler equations serve as a starting point for the building of the PBMs.

Direct application of the methodology of Chicatelli and Hartley, which is related to CFD modeling schemes presented in Hirsch [33, 34], yields a linear, locally-accurate model of the isolator, called the small-perturbation model (SPM). Simulations revealed the model’s inability to accurately capture the flow dynamics, necessitating the development of a nonlinear PBM of the isolator. The size of this model was prohibitive for efficient simulation, so a smaller-order nonlinear model was constructed from the one-dimensional Euler equations. The construction and validation details of each of these models is presented in this chapter.

2.1 CFD Grid and Data Collection

The particular isolator for which model will be constructed has an axisymmetric geometry that is best described as a grid in cylindrical coordinates \((r, \theta, z)\), where \(\hat{r}\) is the radial direction (center-line to wall), \(\hat{\theta}\) is the azimuthal direction, and \(\hat{z}\)
is the streamwise direction (entrance plan to exit plane). The three-dimensional grid constructed for CFD simulation was the one-sixteenth section of the isolator in Figure 2.1, constructed in a rectangular coordinate system and discretized for the simulations. In simulation, mirror symmetry about the $r - z$ plans on the edges of the solution domain allows for construction of the entire flow profiles.

It should be noted that the full CFD grid can be split into three regions: a nozzle region, a constant-area region, and a region with diverging area. The two most-upstream regions are not considered in the modeling efforts described here, as they represent part of the experimental isolator section and not the installed isolator geometry. The section of the isolator to be modeled is from $z = 0.1829$ m to $z = 1.09$ m, with a cross-sectional area that increases linearly from $r = 0.0554$ m to $r = 0.0594$ m. For control, it is assumed that the isolator becomes unstarted when

Figure 2.1: Grid of the $\frac{1}{16}$ slice of the axisymmetric isolator defined for CFD simulation.
the shock wave propagates upstream of the entrance plane of this diverging-area region of the grid (that is, when \( z_{sc} < 0.1829 \) m).

The three-dimensional isolator grid can be simplified for construction of the isolator model under the assumption that the flow is azimuthally-smooth, where the complete flow solution can be constructed from a single azimuthal plane. A specific plane of the grid (fixed \( \theta \)) can be chosen to describe the two-dimensional isolator grid with radial \( (r) \) and streamwise \( (z) \) axes. The grid in Figure 2.1 is reduced to the plane at \( \theta = \pi \) (Figure 2.2, for which the two-dimensional Euler equation-based models in Section 2.2 and the data-based model in Chapter 3 are constructed.)

The two-dimensional grid has \( N_r = 49 \) (non-uniform) intervals in the radial direction, without a point at the center-line \( (r = 0) \), and \( N_z = 149 \) (non-uniform) intervals in the streamwise direction, for a total of \( N_rN_z = 7301 \) grid points. For models constructed from the two-dimensional Euler equations, four flow variables (density,
\( \rho, \) velocity components, \( u_r \) and \( w, \) and internal energy, \( e \) \) are defined at each grid point, so that these models have \( n = 4N_rN_z = 29204 \) states.

Further simplification of the grid can be made by considering a single radial plane (such as the radial plane along the wall, \( r = R(z) \)); this is an acceptable choice for inviscid flow as there is not boundary layer that forms and the flow field is radially smooth as well as azimuthally smooth. (For viscous flow, it would be necessary to consider at least two planes for the model (one in the boundary layer, one in the core flow) in order to accurately capture the flow dynamics.) The one-dimensional grid has \( N_z = 149 \) total points, and three flow variables are defined at each point, so that the model constructed from the one-dimensional Euler equations has \( n = 3N_z = 447 \) states, a reduction by a factor of 67 from the two-dimensional Euler equation-based model. Construction of the smaller-order model is detailed in Section 2.3.

Because the grids for the two models are related, the CFD data collected from simulation of the isolator in Figure 2.1 can be used to construct vectors of initial conditions, as well as flow variable profiles at each time-step, for simulation and validation of each model. AFRL provided a total of ten sets of CFD solutions, in text files compatible with MATLAB\textsuperscript{®}, which could be used during evaluation of each model under study. Three sets contained transient data for the flow field in the isolator under three different multi-step backpressure inputs and were used for construction and validation of the model in Chapter 3. The remaining seven sets of data were collected for use in constructing and tuning the models in this chapter: three sets of data contained transient results for simulation with single-step inputs and four sets contained steady-state solutions for single-step inputs. When each modeling methodology is discussed, more detail will be presented about each specific set of data.
2.2 Two-dimensional Euler equation-based model

The two-dimensional Euler equations are obtained through simplification of the three-dimensional Euler equations, given in equation (2.1), in ‘divergence form.’ This simplification is the starting point for development of the large-scale linear and non-linear isolator models

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0
\]

\[
\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \rho \mathbf{v}) + \nabla p = 0
\]  

\[
\frac{\partial \rho E}{\partial t} + \nabla \cdot \mathbf{v} (\rho E + p) = 0
\]  

(2.1)

The flow variables are density, \( \rho \), velocity, \( \mathbf{v} = u_r \hat{r} + v_\theta \hat{\theta} + w \hat{z} \), pressure, \( p \), and total energy, \( E = e + 0.5(\mathbf{v} \cdot \mathbf{v}) \), and the equations describe the conservation of mass, three-dimensional momentum and energy.

As mentioned in Section 2.1, the shape of the isolator suggests expanding the equations in cylindrical coordinates. It should be noted that the equations can not be further simplified by factoring \( \rho \) out of derivative terms because density is not constant in compressible flow. The expansion of each equation, in cylindrical coordinates \((r, \theta, z)\), are detailed in following sections, from which a set of five partial differential equations are obtained. The two-dimensional Euler equations are found by neglecting the equation for azimuthal momentum and the \( \theta \)-dependent terms in the other four equations.
2.2.1 Continuity Equation

The first line of equation (2.1), the conservation of mass, can be expanded by applying the definition for divergence in cylindrical coordinates:

\[
0 = \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} \\
= \frac{\partial \rho}{\partial t} + \frac{1}{r} \frac{\partial (r \rho v_r)}{\partial r} + \frac{1}{r} \frac{\partial (\rho v_\theta)}{\partial \theta} + \frac{\partial \rho w}{\partial z} \\
= \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_r)}{\partial r} + \frac{1}{r} \frac{\partial (\rho v_\theta)}{\partial \theta} + \frac{\partial \rho w}{\partial z} + \rho u_r
\]  

(2.2)

2.2.2 Momentum Equations

The conservation of momentum equation, the second line in equation (2.1), yields a vector with three components when expanded:

1. The partial derivative taken with respect to time is:

\[
\frac{\partial \rho \mathbf{v}}{\partial t} = \frac{\partial \rho u_r}{\partial t} \hat{r} + \rho u_r \frac{\partial \hat{r}}{\partial t} + \frac{\partial (\rho v_\theta)}{\partial t} \hat{\theta} + \rho v_\theta \frac{\partial \hat{\theta}}{\partial t} + \frac{\partial \rho w}{\partial t} \hat{z} + \rho w \frac{\partial \hat{z}}{\partial t}
\]

\[
= (\frac{\partial \rho u_r}{\partial t} - \rho v_\theta \dot{\theta}) \hat{r} + (\frac{\partial (\rho v_\theta)}{\partial t} - \rho u_r \dot{\theta}) \hat{\theta} + \frac{\partial \rho w}{\partial t} \hat{z}
\]  

(2.3)

where the simplification is made because, for a given cell, \( \theta \) is constant, so its time derivative vanishes.

2. The dyadic product produces a \( 3 \times 3 \) tensor:

\[
\mathbf{v} \otimes \rho \mathbf{v} = \begin{bmatrix}
\rho u_r^2 & \rho u_r v_\theta & \rho u_r w \\
\rho u_r v_\theta & \rho v_\theta^2 & \rho v_\theta w \\
\rho u_r w & \rho v_\theta w & \rho w^2
\end{bmatrix}
\]
The divergence of this tensor yields the vector:

\[
\nabla \cdot \mathbf{v} \otimes \rho \mathbf{v} = \left( \frac{1}{r} \frac{\partial r \rho u_r^2}{\partial r} + \frac{1}{r} \frac{\partial \rho u_r v_\theta}{\partial \theta} + \frac{\partial \rho u_r w}{\partial z} - \frac{\rho v_\theta^2}{r} \right) \hat{r} + \left( \frac{1}{r} \frac{\partial r \rho u_r v_\theta}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta^2}{\partial \theta} + \frac{\partial \rho v_\theta w}{\partial z} + \frac{\rho u_r v_\theta}{r} \right) \hat{\theta} + \left( \frac{1}{r} \frac{\partial r \rho u_r w}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta w}{\partial \theta} + \frac{\partial \rho w^2}{\partial z} \right) \hat{z}
\]

(2.4)

3. The gradient of pressure is expanded in cylindrical coordinates:

\[
\nabla p = \frac{\partial p}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial p}{\partial \theta} \hat{\theta} + \frac{\partial p}{\partial z} \hat{z}
\]

(2.5)

Equations (2.3), (2.4) and (2.5) can be combined to form the momentum equation with \( \hat{r}, \hat{\theta}, \hat{z} \) components:

**r – component:**

\[
0 = \frac{\partial \rho u_r}{\partial t} + \frac{1}{r} \frac{\partial r \rho u_r^2}{\partial r} + \frac{1}{r} \frac{\partial \rho u_r v_\theta}{\partial \theta} + \frac{\partial \rho u_r w}{\partial z} + \frac{\partial p}{\partial r} - \frac{\rho v_\theta^2}{r}
\]

(2.6)

**\( \theta \)–component:**

\[
0 = \frac{\partial \rho v_\theta}{\partial t} + \frac{1}{r} \frac{\partial r \rho u_r v_\theta}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta^2}{\partial \theta} + \frac{\partial \rho v_\theta w}{\partial z} + \frac{\rho u_r v_\theta}{r}
\]

**z – component:**

\[
0 = \frac{\partial \rho w}{\partial t} + \frac{1}{r} \frac{\partial r \rho u_r w}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta w}{\partial \theta} + \frac{\partial \rho w^2}{\partial z} + \frac{\partial p}{\partial z}
\]
2.2.3 Energy Equation

Like the continuity equation, the energy equation can be expanded by applying the identity for divergence:

$$0 = \frac{\partial \rho E}{\partial t} + \nabla \cdot \mathbf{v} (\rho E + p)$$

$$= \frac{\partial \rho E}{\partial t} + \frac{1}{r} \frac{\partial u_r (\rho E + p)}{\partial r} + \frac{1}{r} \frac{\partial v_\theta (\rho E + p)}{\partial \theta} + \frac{\partial w (\rho E + p)}{\partial z} + \frac{u_r (\rho E + p)}{r}$$

(2.7)

2.2.4 Summary of Three-Dimensional Euler Equations

Equations (2.2), (2.6), and (2.7) form the system of partial differential equations summarized here:

$$0 = \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_r}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta}{\partial \theta} + \frac{\partial \rho w}{\partial z} + \frac{\rho u_r}{r}$$

$$0 = \frac{\partial \rho u_r}{\partial t} + \frac{\partial \rho u_r^2 + p}{\partial r} + \frac{1}{r} \frac{\partial \rho u_r v_\theta}{\partial \theta} + \frac{\partial \rho u_r w}{\partial z} + \rho \frac{u_r^2 - v_\theta^2}{r}$$

$$0 = \frac{\partial \rho v_\theta}{\partial t} + \frac{\partial \rho u_r v_\theta}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta^2 + p}{\partial \theta} + \frac{\partial \rho v_\theta w}{\partial z} + 2 \rho \frac{u_r v_\theta}{r}$$

$$0 = \frac{\partial \rho w}{\partial t} + \frac{\partial \rho u_r w}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta w}{\partial \theta} + \frac{\partial \rho w^2 + p}{\partial z}$$

$$0 = \frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_r (\rho E + p)}{\partial r} + \frac{1}{r} \frac{\partial \rho v_\theta (\rho E + p)}{\partial \theta} + \frac{\partial \rho w (\rho E + p)}{\partial z} + \frac{u_r (\rho E + p)}{r}$$

(2.8)

Due to the azimuthal symmetry of flow through the isolator in Figure 2.1, the problem can be simplified by removing dependence on $\theta$ from equation (2.8). The
two-dimensional Euler equations are summarized in equation (2.9).

\[
0 = \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_r}{\partial r} + \frac{\partial \rho w}{\partial z} + \frac{\rho u_r}{r} \\
0 = \frac{\partial \rho u_r}{\partial t} + \frac{\partial \rho u_r^2 + p}{\partial r} + \frac{\partial \rho u_r w}{\partial z} + \frac{\rho u_r^2}{r} \\
0 = \frac{\partial \rho w}{\partial t} + \frac{\partial \rho u_r w}{\partial r} + \frac{\partial \rho w^2 + p}{\partial z} + \frac{\rho u_r w}{r} \\
0 = \frac{\partial \rho E}{\partial t} + \frac{\partial u_r (\rho E + p)}{\partial r} + \frac{\partial w (\rho E + p)}{\partial z} + \frac{u_r (\rho E + p)}{r}
\]  

(2.9)

2.2.5 Derivation of an intermediate nonlinear model

Construction of a state-space model of the system is facilitated by expressing equation (2.9) in a compact form. The state variable vector \( \mathbf{u} = [\rho \; \rho u_r \; \rho w \; \rho E]^T \) and three flux vectors \( f(\mathbf{u}) \), \( h(\mathbf{u}) \) and \( j(\mathbf{u}) \), given in equation (2.10), can be constructed from the partial differential terms in the Euler equations.

\[
f(\mathbf{u}) = \begin{bmatrix} 
\rho u_r \\
\rho u_r^2 + p \\
\rho u_r w \\
u_r (\rho E + p)
\end{bmatrix} \quad h(\mathbf{u}) = \begin{bmatrix} 
\rho w \\
\rho u_r w \\
\rho w^2 + p \\
w (\rho E + p)
\end{bmatrix} \quad j(\mathbf{u}) = \begin{bmatrix} 
\rho u_r \\
\rho u_r^2 \\
\rho u_r w \\
u_r (\rho E + p)
\end{bmatrix}
\]  

(2.10)

Equation (2.9) can then be reduced to a single partial differential equation:

\[
\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial f(\mathbf{u})}{\partial r} + \frac{\partial h(\mathbf{u})}{\partial z} + \frac{1}{r} j(\mathbf{u}) = 0
\]  

(2.11)

The derivatives of the flux functions in equation (2.11) can be transformed by the chain rule to be the product of a Jacobian and a spatial derivative of \( \mathbf{u} \). The resulting form, equation (2.12), lends itself to spatial discretization of the system [35].

\[
\frac{\partial \mathbf{u}}{\partial t} + J_f \frac{\partial \mathbf{u}}{\partial r} + J_h \frac{\partial \mathbf{u}}{\partial z} + \frac{1}{r} J_j \mathbf{u} = 0
\]  

(2.12)
The matrices $J_f$ and $J_h$ are the Jacobians of the flux functions $f(u)$ and $h(u)$ and are given in terms of flow variables in equations (2.13) and (2.14).

$$J_f = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-u_r^2 + \frac{\gamma-1}{2}v \cdot v & (3-\gamma)u_r & -(\gamma-1)w & (\gamma-1) \\
u_r w & w & u_r & 0 \\
-u_r(\gamma E - (\gamma - 1)v \cdot v) & \gamma E - (\gamma - 1)u_r^2 - \frac{\gamma-1}{2}v \cdot v & -(\gamma-1)u_r w & \gamma u_r
\end{bmatrix}$$  \hspace{1cm} (2.13)

$$J_h = \begin{bmatrix}
0 & 0 & 1 & 0 \\
u_r w & w & u_r & 0 \\
-w^2 + \frac{\gamma-1}{2}v \cdot v & -(\gamma-1)u_r & (3-\gamma)w & (\gamma-1) \\
w(\gamma E - (\gamma - 1)v \cdot v) & -(\gamma-1)u_r w & \gamma E - (\gamma - 1)w^2 - \frac{\gamma-1}{2}v \cdot v & \gamma w
\end{bmatrix}$$  \hspace{1cm} (2.14)

Although the flux vector $j(u)$ does not appear in a spatial partial derivative, it can be expressed as $j(u) = J_j u$ though application of Euler’s theorem of homogeneous functions, where the Jacobian $J_j$ is given in equation (2.15).

$$J_j = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-u_r^2 & 2u_r & 0 & 0 \\
u_r w & w & u_r & 0 \\
-u_r(\gamma E - (\gamma - 1)v \cdot v) & \gamma E - (\gamma - 1)u_r^2 - (\gamma - 1)\frac{\gamma-1}{2}v \cdot v & -(\gamma-1)u_r w & \gamma u_r
\end{bmatrix}$$  \hspace{1cm} (2.15)

### 2.2.6 Spatial discretization of the Euler equations

The Euler equations govern the flow at each point in the grid in Figure 2.2 since they describe the flow throughout the isolator. The nonlinear model is constructed by assigning an index pair $(i, k)$ to each grid point and spatially discretizing equation (2.12) at each point. Index $k$ is 1 at the entrance of the isolator and $N_z$ at
the exit, and index \( i \) is 1 at the center-line and \( N_r \) at the wall. A discussion of the importance of defining \( i \) correctly will be presented in Section 2.2.8, where it will be shown that reversing the order of the radial indexing caused the linearized model to be unstable unless constructed assuming a rectangular coordinate system.

The \( n \)-dimensional state vector \( x \) in equation (2.16) is constructed by stacking the vectors \( u_{i,k} \) in the order: \( (i = 1(k = 1...N_z)...N_r) \):

\[
x = \begin{bmatrix}
u_{1,1} \\
u_{1,2} \\
\vdots \\
u_{1,N_z} \\
u_{2,1} \\
u_{2,2} \\
\vdots \\
u_{N_r,N_z}
\end{bmatrix} \in \mathbb{R}^{4N_rN_z} = \mathbb{R}^n
\] (2.16)

The interconnections between neighboring grid points can be described through spatial discretization of equation (2.12); as in Ref. 29, flux-splitting is implemented. Introduced by Steger and Warming in Ref. 36, flux splitting preserves the direction that each flow characteristic travels by separating the positive and negative characteristics during discretization. This maintains stability in the discretized system when the flow is subsonic.

Flux-splitting requires defining split-Jacobians \( J^\pm \) from the eigenvector matrices of each Jacobian and a diagonal matrix containing the positive or negative eigenvalues. The eigenvalues for the Jacobians are \( \lambda(J_f) = \{u_r, u_r, u_r + c, u_r - c\} \) and \( \lambda(J_h) = \{w, w, w + c, w - c\} \), where \( c \) is the sonic velocity. The repeated eigenvalues \( u_r \) and \( w \) have equal algebraic and geometric multiplicities, so that four non-generalized
eigenvectors can be defined for each Jacobian (equations (2.17) and (2.18)). (The free parameters were chosen in such a way to have Ref. 28 as a point of comparison.)

\[
K_f = \begin{bmatrix}
1 & 0 & \frac{\rho}{2c} & -\frac{\rho}{2c} \\
u_r & 0 & \frac{\rho}{2c}(u_r + c) & -\frac{\rho}{2c}(u_r - c) \\
w & \rho & \frac{\rho w}{2c} & -\frac{\rho w}{2c} \\
\frac{\sqrt{\gamma}}{2} & \rho w & \frac{\rho}{2c}(\frac{\sqrt{\gamma}}{2} + u_r c + \frac{c^2}{\gamma - 1}) & -\frac{\rho}{2c}(\frac{\sqrt{\gamma}}{2} - u_r c + \frac{c^2}{\gamma - 1})
\end{bmatrix}
\]

\(2.17\)

\[
K_h = \begin{bmatrix}
0 & 1 & \frac{\rho}{2c} & -\frac{\rho}{2c} \\
\rho & u_r & \frac{\rho u_r}{2c} & -\frac{\rho u_r}{2c} \\
0 & w & \frac{\rho}{2c}(w + c) & -\frac{\rho}{2c}(w - c) \\
\rho u_r & \frac{\sqrt{\gamma}}{2} & \frac{\rho}{2c}(\frac{\sqrt{\gamma}}{2} + wc + \frac{c^2}{\gamma - 1}) & -\frac{\rho}{2c}(\frac{\sqrt{\gamma}}{2} - wc + \frac{c^2}{\gamma - 1})
\end{bmatrix}
\]

\(2.18\)

Following Steger and Warming [36], the matrices \(A_{f,h}^\pm\) are defined to satisfy \(A_{f,h} = A_{f,h}^+ + A_{f,h}^-\) (equation (2.19)), separating the positive and negative eigenvalues, and the flux-split Jacobians \(J_{f,h}^\pm = K_{f,h} A_{f,h}^\pm K_{f,h}^{-1}\), where \(K_{f,h}^{-1}\) are given in equation (2.20).

By definition, \(J_{f,h}^\pm\) satisfies \(J_{f,h} = J_{f,h}^+ + J_{f,h}^-\).

\[
A_{f}^+ = \begin{bmatrix}
\frac{1}{2}(u_r \pm |u_r|) & 0 & 0 & 0 \\
0 & \frac{1}{2}(u_r \pm |u_r|) & 0 & 0 \\
0 & 0 & \frac{1}{2}(u_r \pm c \pm |u_r + c|) & 0 \\
0 & 0 & 0 & \frac{1}{2}(u_r - c \pm |u_r - c|)
\end{bmatrix}
\]

\(2.19\)

\[
A_{h}^+ = \begin{bmatrix}
\frac{1}{2}(w \pm |w|) & 0 & 0 & 0 \\
0 & \frac{1}{2}(w \pm |w|) & 0 & 0 \\
0 & 0 & \frac{1}{2}(w + c \pm |w + c|) & 0 \\
0 & 0 & 0 & \frac{1}{2}(w - c \pm |w - c|)
\end{bmatrix}
\]
Given the definitions of $J_f^+$ and $J_h^+$, governing equations for the $(N_r - 2)(N_z - 2)$ points on the interior of the grid are found by using a backwards-difference to discretize terms related to the positive characteristics ($J_{f,h}^+$) and a forward-difference to discretize those related to the negative characteristics ($J_{f,h}^-$):

\[
\frac{\partial u}{\partial t} + J_f^+ \frac{u_{i,k} - u_{i-1,k}}{r_{i,k} - r_{i-1,k}} + J_f^- \frac{u_{i+1,k} - u_{i,k}}{r_{i+1,k} - r_{i,k}} + J_h^+ \frac{u_{i,k} - u_{i,k-1}}{z_{i,k} - z_{i,k-1}} + J_h^- \frac{u_{i,k+1} - u_{i,k}}{z_{i,k+1} - z_{i,k}} = 0
\]

(2.21)

For grid points at the boundaries, equation (2.21) must be modified so the finite-difference calculations do not depend on points outside the computational domain. At the $k = N_z$ plane, the boundary condition functions must contain the system input (backpressure, $p_{in}$).
Modifications for Boundary Cells – Radial Direction

In the $\hat{r}$ direction, modifications must be made for the boundaries at $i = 1$ and at $i = N_r$, the center-line and wall. In both cases, only one finite-difference in $r$ can be implemented in equation (2.21): the forward-difference (multiplied by $J_f^-$) when $i = 1$, and the backwards-difference (multiplied by $J_f^+$) when $i = N_r$.

Modifications for Boundary Cells – Streamwise Direction

In the $\hat{z}$ direction, the boundary condition treatment described by Chicatelli and Hartley [28,29], which is based on compatibility relations and time-differenced boundary conditions as detailed by Hirsch [33,34], is implemented here. This requires defining boundary condition functions $B_{bc}$ to replace streamwise characteristics entering the computational domain at the inflow and outflow planes.

Since inviscid flow is considered, velocity is assumed to be supersonic across the entire entrance plane of the isolator. At the exit plane, however, it will be necessary to implement the boundary condition treatment both for subsonic and supersonic flow, because a scramjet engine may have subsonic flow at the exit plane when operating in low-Mach number regions of the flight envelope. Although Chicatelli and Hartley do not treat the case of a downstream input with supersonic flow, the boundary condition treatment here is adapted from their case of an upstream input with supersonic flow [28].

In general, this boundary condition treatment involves partitioning $K_{h}^{-1}$ (equation (2.20)) by grouping separately the rows associated with the numerical characteristics (which depend on data inside the grid) and physical characteristics (which depend on data outside the grid). The sign on the eigenvalues of $J_h$ determine the direction of the characteristic: positive eigenvalues describe characteristics that travel downstream while negative eigenvalues describe those that travel upstream. At the
entrance plane, positive characteristics require information from outside the computational grid and are therefore physical characteristics defined by boundary conditions. At the exit plane, the physical characteristics are described by negative eigenvalues. The number of physical and numerical characteristics at each grid point is dependent on the speed of flow.

A coordinate change to transform the system from state variables \( u \) to characteristic variables \( w \) is defined by relating the time derivatives of these two coordinates by \( K_h \). In the characteristic variable coordinate system, states associated with physical characteristics are replaced with \( B_{bc} = 0 \); time-differencing of the boundary condition yields the expression \( \partial B_{bc}/\partial t = -(\Delta T)^{-1} B_{bc} \), where \( \Delta T \) is the time-step of the simulation \([33,34]\). Transforming back to the state variables \( u \) yields the expression

\[
\frac{\partial u}{\partial t} + \left[ \frac{(K_h^{-1})_N}{\partial B_{bc}} \right]^{-1} \left[ \begin{array}{c} (K_h^{-1})_N \\ 0 \end{array} \right] \frac{\partial u}{\partial r} + \left[ \frac{(K_h^{-1})_N}{\partial B_{bc}} \right]^{-1} \left[ \begin{array}{c} (K_h^{-1})_N \\ 0 \end{array} \right] \frac{\partial u}{\partial z} = \frac{1}{r} \left[ \frac{(K_h^{-1})_N}{\partial B_{bc}} \right]^{-1} \left[ \begin{array}{c} (K_h^{-1})_N \\ 0 \end{array} \right] J_h \frac{\partial u}{\partial z}
\]  

(2.22)

where \( (K_h^{-1})_N \) is the partition of \( K_h^{-1} \) associated with the numerical characteristics.

At the isolator entrance plane, where the flow is supersonic, \( (K_h^{-1})_N \) is the empty matrix because \( J_h \) has no negative eigenvalues and it is necessary to define four boundary conditions at the entrance plane: \( B_{bc}^{in}(u) = [\rho \ \rho u \ \rho w \ \rho E]^T \). With these substitutions, equation (2.22) simplifies to

\[
\frac{\partial u}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u)
\]  

(2.23)

which will be further simplified when constructing the linearized and nonlinear affine models.
Two separate boundary conditions are defined at the isolator exit plane. When the streamwise flow is supersonic, there are no numerical characteristics and the boundary conditions are defined by

\[
B_{\text{bc, sup}}^{\text{out}}(u, p_{\text{in}}) = \begin{bmatrix}
\rho \\
\rho u \\
p - p_{\text{in}} \\
\rho E
\end{bmatrix}
\]  

(2.24)

The system input, \( p_{\text{in}} \), is in the third characteristic, following the treatment in Ref. 29. Because \((K_h^{-1})_N\) is empty, the governing equations for these grid points have the form in equation (2.23), with \(B_{\text{bc, sup}}^{\text{out}}(u, p_{\text{in}})\) replacing \(B_{\text{bc}}^{\text{in}}(u)\).

For subsonic flow, there is only one physical characteristic and the boundary condition is defined by

\[
B_{\text{bc, sub}}^{\text{out}}(u, p_{\text{in}}) = p - p_{\text{in}}
\]  

(2.25)

The three numerical characteristics are associated with the positive eigenvalues of \(J_h\), for which the right vectors are the first three rows of \(K_h^{-1}\). These rows comprise the matrix partition \((K_h^{-1})_N\) which, along with the definition of \(B_{\text{bc, sub}}^{\text{out}}(u, p_{\text{in}})\), allows for definition of matrices \(L\) and \(L_B\) (equations (2.26) and (2.27)), where

\[
\psi = \left\{ \frac{v \cdot v}{2} - wc + \frac{c^2}{\gamma - 1} \right\}
\] in \(L\):
\[
L = \left[ \frac{(K^{-1}_h)_N}{\partial B_{bc,sub}^{out}} \right]^{-1} \left[ \begin{array}{c}
\frac{(K^{-1}_h)_N}{\partial u} \\
0
\end{array} \right] \\
= \left[ 
\begin{array}{cccc}
1 - \frac{\gamma-1}{2}v \cdot v & \frac{(\gamma-1)u_r}{c^2} & \frac{(\gamma-1)w}{c^2} & -\frac{\gamma-1}{c^2} \\
-\frac{(\gamma-1)v \cdot v}{2c^2}ur & \frac{(\gamma-1)u_r^2}{c^2} + 1 & \frac{(\gamma-1)uw}{c^2} & \frac{(\gamma-1)uw}{c^2} \\
-\frac{(\gamma-1)v \cdot v}{2c^2}(w - c) & -\frac{(\gamma-1)u_r(w - c)}{c^2} & \frac{(\gamma-1)uw}{c^2} + 1 & -\frac{\gamma-1}{c^2}(w - c) \\
-\frac{(\gamma-1)v \cdot v}{2c^2}\psi & \frac{(\gamma-1)u_r\psi}{c^2} & \frac{(\gamma-1)w\psi}{c^2} & -\frac{\gamma-1}{c^2}\psi
\end{array} \right]
\]

\[
L_B = \left[ \frac{(K^{-1}_h)_N}{\partial B_{bc,sub}^{out}} \right]^{-1} \left[ \begin{array}{c}
0 \\
-\frac{1}{\Delta T} \rho B_{bc,sub}^{out}(u, p_{in})
\end{array} \right] \\
= \left[ 
\begin{array}{cccc}
-\frac{u_r}{\rho} & \frac{1}{\rho} & 0 & 0 \\
1 - \frac{\gamma-1}{2}v \cdot v & \frac{(\gamma-1)u_r}{c^2} & \frac{(\gamma-1)w}{c^2} & -\frac{\gamma-1}{c^2} \\
\frac{1}{\rho c} \left\{ \frac{\gamma-1}{2}v \cdot v - wc \right\} & -\frac{(\gamma-1)u_r}{\rho c} & \frac{1}{\rho c} \left\{ c - (\gamma - 1)w \right\} & \frac{\gamma-1}{\rho c} \\
\frac{\gamma-1}{2}v \cdot v & -(\gamma - 1)u_r & -(\gamma - 1)w & (\gamma - 1)
\end{array} \right]^{-1} \\
\times \left[ 
\begin{array}{c}
0 \\
0 \\
0 \\
-\frac{1}{\Delta T} (p - p_{in})
\end{array} \right]
\]

(2.26)

The governing equations at the exit plane can be compactly written as in equation (2.28). Note that flux-splitting is not implemented here in the \( \hat{z} \)-direction as only the backwards-difference is used to approximate the partial derivative with respect to \( z \). The term with the partial derivative with respect to \( r \) is discretized through implementation of flux-splitting or by applying the boundary condition treatment, depending on \( i \)

\[
\frac{\partial u}{\partial t} + LJf \frac{\partial u}{\partial r} + LJh \frac{\partial u}{\partial z} + \frac{1}{r} L_{J_j} u = L_B
\]

(2.28)
Like equation (2.23), expansion of the nonlinear terms in equation (2.28) yields linear
and nonlinear, affine models of the isolator.

2.2.7 Constructing the Intermediate Nonlinear Model

The state vector $x$ defined in equation (2.16) and the definitions of the governing
equations (2.21), (2.23), and (2.28) can be used to construct an intermediate form of
the isolator model:

$$\dot{x} = \bar{A}(x)x + \bar{B}(x, p_{in})$$  \hspace{1cm} (2.29)

Linearization of this model produces the SPM described in Section 2.2.8 and expansion
yields an affine form of the model, presented in Section 2.2.9.

Expansion of the governing equations can be simplified by employing the notation
for finite differences given by:

$$\begin{align*}
\Delta^b(\cdot) &= (\cdot)_{i,k} - (\cdot)_{i-1,k} & \Delta^f(\cdot) &= (\cdot)_{i+1,k} - (\cdot)_{i,k} \\
\Delta^b(\cdot) &= (\cdot)_{i,k} - (\cdot)_{i,k-1} & \Delta^f(\cdot) &= (\cdot)_{i,k+1} - (\cdot)_{i,k}
\end{align*}$$  \hspace{1cm} (2.30)

The expanded governing equations are given in Table 2.1, where each boundary con-
dition treatment is shown. Here, case 8 replaces the expressions in cases 3, 5, and 7
when the flow is supersonic at the exit plane, as indicated in the table. The majority
of grid points are governed by the form in case 4:

Expansion of the expressions in Table 2.1 yield the state equations in equa-
tion (2.31), where subscripts have been dropped from the states and Jacobians as-
associated with the cell identified by the index pair $(i, k)$, from which an intermediate
matrix $\bar{A}(x)$ can be constructed. The non-zero element in each block row of this ma-
trix are given by the coefficients on $u_{i,k}$, $u_{i\pm1,k}$, and $u_{i,k\pm1}$ in each governing equation.
Terms on the RHS of the expressions in Table 2.1 can be gathered to form a nonlinear
Table 2.1: Explicit definition of the equations applied at each point in the CFD grid. Indices \( i \) and \( k \) correspond to the interval in the radial and streamwise directions to which each equation is applied, and the difference notation is given in equation (2.30). Note that where subscripts are dropped, it is assumed they are \((i, k)\).

<table>
<thead>
<tr>
<th>case</th>
<th>( i )</th>
<th>( k )</th>
<th>equation form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \in (1, N_r) )</td>
<td>1</td>
<td>( \frac{\partial u}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc}^n}{\partial u} \right)^{-1} B_{bc}^m(u) )</td>
</tr>
<tr>
<td>2</td>
<td>( 1 )</td>
<td>( \in (2, N_z - 1) )</td>
<td>( \frac{\partial u}{\partial t} + \frac{J_f}{\Delta r} \Delta^r \Delta^f u + \frac{J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} J u = 0 )</td>
</tr>
<tr>
<td>3</td>
<td>( 1 )</td>
<td>( N_z ) (subsonic)</td>
<td>( \frac{\partial u}{\partial t} + \frac{L J_f}{\Delta r} \Delta^r \Delta^f u + \frac{L J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} L J u = L_B(u, p_{in}) )</td>
</tr>
<tr>
<td>4</td>
<td>( \in (2, N_r - 1) )</td>
<td>( \in (2, N_z - 1) )</td>
<td>( \frac{\partial u}{\partial t} + \frac{J_f}{\Delta r} \Delta^r \Delta^f u + \frac{J_f}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} J u = 0 )</td>
</tr>
<tr>
<td>5</td>
<td>( \in (2, N_r - 1) )</td>
<td>( \in (2, N_z - 1) )</td>
<td>( \frac{\partial u}{\partial t} + \frac{L J_f}{\Delta r} \Delta^r \Delta^f u + \frac{L J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} L J u = L_B(u, p_{in}) )</td>
</tr>
<tr>
<td>6</td>
<td>( N_r )</td>
<td>( \in (2, N_z - 1) )</td>
<td>( \frac{\partial u}{\partial t} + \frac{J_f}{\Delta r} \Delta^r \Delta^f u + \frac{J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} J u = 0 )</td>
</tr>
<tr>
<td>7</td>
<td>( N_r )</td>
<td>( N_z ) (subsonic)</td>
<td>( \frac{\partial u}{\partial t} + \frac{L J_f}{\Delta r} \Delta^r \Delta^f u + \frac{L J^+}{\Delta x} \Delta^b \Delta^{kb} u + \frac{1}{r} L J u = L_B(u, p_{in}) )</td>
</tr>
<tr>
<td>8</td>
<td>( \in (2, N_r) )</td>
<td>( N_z ) (supersonic)</td>
<td>( \frac{\partial u}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc, sup}^n}{\partial u} \right)^{-1} B_{bc, sup}^m(u, p_{in}) )</td>
</tr>
</tbody>
</table>
matrix function $\vec{B}(x, p_{in})$, from which $A(x)$ and $B(x)$ can be constructed.

\[

t = 1 \quad \begin{array}{ll}
    k = 1 & \frac{\partial u}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) \\
    k = 2 & \frac{\partial u}{\partial t} = -\frac{j_f}{\Delta \nu_f} u_{1,2} + \frac{j_{h}}{\Delta \nu_{h}} u_{1,1} - \frac{j_{f}}{\Delta \nu_{f}} u_{1,1} \\
    & + \left( \frac{j_{f}}{\Delta \nu_{f}} - \frac{j_{h}}{\Delta \nu_{h}} + \frac{j_{f}}{\Delta \nu_{f}} - \frac{1}{\gamma J_{f}} \right) u \\
    \vdots
\end{array}
\]

\[

t = N_z \quad \begin{array}{ll}
    k = 2 & \frac{\partial u}{\partial t} = \frac{L J_{f}}{\Delta \nu_f} u_{2, N_z} + \frac{L J_{h}}{\Delta \nu_{h}} u_{1, N_z - 1} \\
    & + \left( \frac{L J_{f}}{\Delta \nu_f} - \frac{L J_{h}}{\Delta \nu_{h}} - \frac{1}{\gamma L J_{f}} \right) u + L_B(u, p_{in}) \\
    \vdots
\end{array}
\]

\[

t = N_r \quad \begin{array}{ll}
    k = 2 & \frac{\partial u}{\partial t} = \frac{L J_{f}}{\Delta \nu_f} u_{N_r - 1, 1} + \frac{L J_{h}}{\Delta \nu_{h}} u_{N_r, 1} - \frac{L J_{h}}{\Delta \nu_{h}} u_{N_r, 3} \\
    & + \left( \frac{L J_{f}}{\Delta \nu_f} - \frac{L J_{h}}{\Delta \nu_{h}} + \frac{L J_{h}}{\Delta \nu_{h}} - \frac{1}{\gamma J_{f}} \right) u \\
    \vdots
\end{array}
\]

Note that the expressions in equation (2.31) assume subsonic streamwise flow at the isolator exit. In the case of a scramjet, the core flow will be supersonic, and in this case the governing equations are replaced by:

\[
\frac{\partial u}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc, sup}^{out}}{\partial u} \right)^{-1} B_{bc, sup}^{out}(u, p_{in})
\]
When constructing the model, the streamwise velocity component at each point along the exit plane is checked so that the ‘correct’ form of the governing equations is implemented at that point.

**Building the intermediate matrix \( \bar{A}(x) \)**

The structure of the \( 4N_rN_z \)-by-\( 4N_rN_z \) matrix \( \bar{A}(x) \) forms the base of the state matrix in both the linearized and nonlinear affine models and can be built using the submatrices defined in equations (2.33) and (2.34) to (2.39), where velocity dependence is indicated in the subscripts. Here 0 is the 4-by-4 zero matrix and the appropriate forward- or backwards-differences should be used in place of \( \Delta r \) in equation (2.33). Note also that indices on the Jacobians have been dropped in equations (2.34) to (2.39).

\[
J_{F,\text{sub},i}^\pm(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \pm \frac{J_{L,(i,2)}^\pm}{\Delta r} & 0 & \ldots & 0 & 0 \\
0 & 0 & \pm \frac{J_{L,(i,3)}^\pm}{\Delta r} & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & \pm \frac{J_{L,(i,N_z-1)}^\pm}{\Delta r} & 0 \\
0 & 0 & 0 & \ldots & 0 & \pm \frac{L_{J_{L,(i,N_z)}}^\pm}{\Delta r}
\end{bmatrix}
\]

\[
J_{F,\text{sup},i}^\pm(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \pm \frac{J_{L,(i,2)}^\pm}{\Delta r} & 0 & \ldots & 0 & 0 \\
0 & 0 & \pm \frac{J_{L,(i,3)}^\pm}{\Delta r} & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \cdots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & \pm \frac{J_{L,(i,N_z-1)}^\pm}{\Delta r} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}
\]
\[ \tilde{A}_{1, \text{sub}}(x) = \begin{bmatrix}
0 & 0 & \ldots & 0 \\
\frac{J^+}{\Delta^{x/z}} & \frac{J^+}{\Delta^{x/z}} - \frac{J^-}{\Delta^{x/z}} - \frac{1}{r}J_3 & 0 & \ldots \\
0 & \frac{J^+}{\Delta^{x/z}} & \frac{J^+}{\Delta^{x/z}} + \frac{J^-}{\Delta^{x/z}} + \frac{1}{r}J_3 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 \\
\ldots & 0 & \ldots & 0 & \ldots & 0 \\
\ldots & 0 & \ldots & 0 & \ldots & 0 \\
\ldots & \vdots & \vdots & \vdots & \ddots & \ddots \\
\ldots & 0 & \ldots & \frac{J^+}{\Delta^{x/z}} & \frac{J^+}{\Delta^{x/z}} + \frac{J^-}{\Delta^{x/z}} - \frac{1}{r}J_3 & \ldots \\
\ldots & 0 & \ldots & 0 & \frac{J^+}{\Delta^{x/z}} & \frac{J^+}{\Delta^{x/z}} - \frac{1}{r}LJ_3 \\
\end{bmatrix} \]
\[
A_{1,sup}(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & \ldots \\
\frac{J^+}{\Delta kz} & \frac{J^-}{\Delta k z} - \frac{J^+}{\Delta k z} + \frac{J^-}{\Delta k z} - \frac{1}{r} J_j & 0 & 0 & \ldots \\
0 & \frac{J^+}{\Delta k z} & \frac{J^-}{\Delta k z} - \frac{J^+}{\Delta k z} + \frac{J^-}{\Delta k z} - \frac{1}{r} J_j & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & \ldots \\
\ldots & 0 & \ldots & 0 & \ldots \\
\ldots & 0 & \ldots & 0 & \ldots \\
\ldots & 0 & \ldots & 0 & \ldots \\
\ldots & \vdots & \vdots & \vdots & \ddots \\
\ldots & 0 & \ldots & 0 & 0
\end{bmatrix}
\]

(2.35)
\[ \tilde{A}_{N_r, \text{sub}}(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & \cdots \\
\frac{J^+}{\Delta x^r_z} & -\frac{J^+}{\Delta x^r_z} - \frac{J^-}{\Delta x^r_z} + \frac{1}{c} J_j & 0 & 0 & \cdots \\
0 & \frac{J^+}{\Delta x^r_z} & -\frac{J^+}{\Delta x^r_z} - \frac{J^-}{\Delta x^r_z} - \frac{1}{c} J_j & 0 & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & \cdots
\end{bmatrix} \] (2.36)
\[ \bar{A}_{N',sup}(x) = \begin{bmatrix}
0 & -J^+_z \Delta z - J^+_y \Delta y - \frac{1}{\tau}J_y & -J^-_x \Delta x \Delta z & 0 & \cdots \\
J^+_z \Delta z & 0 & J^+_y \Delta y - \frac{1}{\tau}J_y & -J^-_x \Delta x \Delta z & 0 & \cdots \\
0 & J^-_x \Delta x \Delta z & -J^+_y \Delta y - J^-_x \Delta x \Delta z - \frac{1}{\tau}J_y & J^-_y \Delta y \Delta z & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & J^+_z \Delta z & -J^+_y \Delta y - J^-_x \Delta x \Delta z - \frac{1}{\tau}J_y & J^-_y \Delta y \Delta z & \cdots \\
0 & 0 & J^-_x \Delta x \Delta z & J^-_y \Delta y \Delta z & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\end{bmatrix} \] 

(2.37)
$$\tilde{A}_{i,\text{sub}}(x) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \ldots \\ \frac{J^+}{\Delta v_{r}} - \frac{J^-}{\Delta v_{r}} - \frac{J^+}{\Delta v_{z}} + \frac{J^-}{\Delta v_{z}} - \frac{J_3}{r} & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{J}{\Delta v_{r}} - \frac{J^-}{\Delta v_{r}} - \frac{J^+}{\Delta v_{z}} + \frac{J^-}{\Delta v_{z}} - \frac{J_3}{r} & 0 & 0 & 0 & 0 & \ldots \\ \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{z}} - \frac{L_J}{r} & \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{z}} - \frac{L_J}{r} & \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{r}} - \frac{L_J}{\Delta v_{z}} - \frac{L_J}{r} & \ldots \end{bmatrix}$$

(2.38)
\[
\hat{A}_{i,\text{sup}}(x) = \begin{bmatrix}
0 & -\frac{J_+}{\Delta v_r} + \frac{J_r}{\Delta v_r} - \frac{J_+}{\Delta v_z} + \frac{1}{r} J_j - \frac{J_r}{\Delta v_z} & 0 & \cdots \\
\frac{J_+}{\Delta v_r} & 0 & -\frac{J_r}{\Delta v_r} + \frac{J_r}{\Delta v_z} + \frac{1}{r} J_j - \frac{J_r}{\Delta v_z} & 0 & \cdots \\
0 & \frac{J_+}{\Delta v_r} & 0 & -\frac{J_r}{\Delta v_r} + \frac{J_r}{\Delta v_z} + \frac{1}{r} J_j - \frac{J_r}{\Delta v_z} & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{J_+}{\Delta v_r} & -\frac{J_+}{\Delta v_r} + \frac{J_r}{\Delta v_r} - \frac{J_+}{\Delta v_z} + \frac{1}{r} J_j - \frac{J_r}{\Delta v_z} & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\end{bmatrix}
\]
Given these submatrices, the intermediate matrix $\bar{A}(x)$ is constructed with the structure given in equation (2.40), where the flow conditions at each boundary grid point determine the structure of each submatrix. The one-sided finite-differences used for discretization lead to the block tridiagonal structure of the matrix

$$A = \begin{bmatrix}
\bar{A}_1 & J_{F,1}^- & \zeta & \zeta & \ldots & \zeta \\
J_{F,2}^+ & \bar{A}_2 & J_{F,2}^- & \zeta & \ldots & \zeta \\
\zeta & J_{F,3}^+ & \bar{A}_3 & J_{F,3}^- & \ldots & \zeta \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\zeta & \ldots & \zeta & J_{F,N_r-1}^+ & \bar{A}_{N_r-1}^- & J_{F,N_r-1}^- \\
\zeta & \ldots & \ldots & \zeta & J_{F,N_r}^+ & \bar{A}_{N_r} \\
\end{bmatrix} \tag{2.40}$$

**Building the nonlinear boundary condition vector $\bar{B}(x,u)$**

Boundary condition-dependent nonlinear terms are collected to form the $4N_rN_z$-by-1 vector $B(x,p_{in})$, which will contribute to the construction of both $A(x)$ and $B(x)$ after expansion. Defining $Z$ as the 4-by-1 zero vector, the submatrices in equations (2.41) and (2.42) can be constructed, where subscripts on $u_i,k$ have been dropped:

$$B_{i,\text{sub}}(x,p_{in}) = \begin{bmatrix}
-\frac{1}{\Delta T} \left( \frac{\partial B_{in}^a}{\partial u} \right)^{-1} B_{bc}^{in}(u) \\
Z \\
\vdots \\
Z \\
L_B(u,p_{in})
\end{bmatrix} \tag{2.41}$$
These $N_r$ vectors can be stacked to form $\bar{B}(x, p_{in})$, as follows

$$\bar{B}(x, p_{in}) = \begin{bmatrix} B_1(x, p_{in}) \\ B_2(x, p_{in}) \\ \vdots \\ B_{N_r}(x, p_{in}) \end{bmatrix}$$

(2.43)

where the submatrices are chosen based on the streamwise component at the exit plane.

From the matrices defined in equations (2.40) and (2.43), the nonlinear form can be given in equation (2.29). The isolator model will be completed through definition of an output equation, $y = C(x)x + Dp_{in}$, where $C(x) = C$ in the linearized model. The output equations for the linearized model will be presented in Section 2.2.8 and for the nonlinear model in Section 2.2.9.

### 2.2.8 Linearized model

A locally-accurate small-perturbation model of the inlet/isolator subsystem can be constructed through linearization of equation (2.29) about a steady-state CFD solution, following the methodology developed by Chicatelli and Hartley [28, 29]. A nonlinear output function is defined to complete the characterization of the model.
and a linear time-invariant system (LTI) can be found through linearization of the state and output equations.

**Building the (Nonlinear) Output Function**

The matrices $C$ and $D$ can be constructed through linearization of an output function. For the linearized isolator model, the output at a grid point $(i, k)$ is defined as $y_{i,k} = [u_r \ w \ p \ T]^T_{i,k}$, where pressure $p = (\gamma - 1)\rho(E - 0.5v \cdot v)$ and temperature is found from the gas equation ($T = p/(\rho R)$, where $R$ is the universal gas constant). The output is measured at each point in the planes at each quarter-length of the isolator, plus the entrance and exit planes (a total of $20N_r$ inputs). The quarter-length locations in the isolator with length $L$ are defined by the indices $k$ corresponding to cross-sections at $z = nL/4$, where $n$ is an integer between 0 and 4 (inclusive).

For grid points upstream of the isolator exit (that is, those points with $k \neq N_z$), output is a function of the state variables, $y_{i,k} = k_{i,k}(u_{i,k})$. For points in the exit plane, the output is the sum of a state-dependent function $k'_{i,N_z}(u_{i,N_z})$ and a feedthrough function $l_{i,N_z}(p_{in,N_z})$, where $p_{in,N_z}$ is the backpressure imposed on the exit plane (assumed constant across the entire plane):

$$y = k'(u) + l(p_{in})$$  \hspace{1cm} (2.44)

where

$$k'(u) = \begin{bmatrix} u_r \\ w \\ 0 \\ T \end{bmatrix} \quad \text{and} \quad l(p_{in}) = \begin{bmatrix} 0 \\ 0 \\ p_{in} \\ 0 \end{bmatrix}$$

40
The output vector, $y$, can be formed by stacking the $y_{i,k}$ with the same order of indices used in constructing $x$ (equation (2.16)). Let $Z$ be the 4-by-1 zero vector; then, the $20N_r$-by-1 output vector is:

$$
y = \begin{bmatrix} y_{1,1} \\ y_{1,k_2} \\ y_{1,k_3} \\ y_{1,k_4} \\ y_{1,N_z} \\ y_{2,1} \\ \vdots \\ y_{N_r,N_z} \end{bmatrix} = \begin{bmatrix} k_{1,1}(u_{1,1}) \\ k_{1,k_2}(u_{1,k_2}) \\ k_{1,k_3}(u_{1,k_3}) \\ k_{1,k_4}(u_{1,k_4}) \\ k'_{1,N_z}(u_{1,N_z}) \\ k_{2,1}(u_{2,1}) \\ \vdots \\ k'_{N_r,N_z}(u_{N_r,N_z}) \end{bmatrix} + \begin{bmatrix} Z \\ Z \\ Z \\ Z \\ l(p_{in,(1,N_z)}) \\ Z \\ \vdots \\ l(p_{in,(N_r,N_z)}) \end{bmatrix} = k(x) + l(p_{in}) \quad (2.45)$$

Equations (2.29) (state equations) and (2.45) (output equations) complete a nonlinear characterization of the isolator system:

$$\begin{aligned}
\dot{x} &= \bar{A}(x)x + \bar{B}(x,p_{in}) \\
y &= k(x) + l(p_{in}) 
\end{aligned} \quad (2.46)$$

**Linearizing the Nonlinear System**

A linear model capturing the transient response due to small input perturbations can be found through linearization of equation (2.46) about a steady-state condition using Taylor series approximations. The system matrices $A$, $B$, $C$, and $D$ that characterize the model can be defined through the linearization. Here, the state, input and output of the linear system are *perturbations* from steady-state. The state matrix $A$ and input matrix, $B$ can be constructed through linearization of the expansion of the nonlinear state equation (equation (2.29)) about the steady-state values of state
\[ \begin{align*}
\dot{x} + \delta x &= \bar{A}(\bar{x} + \delta x)(\bar{x} + \delta x) + \bar{B}(\bar{x} + \delta x, \bar{p}_{in} + \delta p_{in}) \\
&= \bar{A}(\bar{x})\bar{x} + \bar{B}(\bar{x}, \bar{p}_{in}) + \left[ \frac{\partial}{\partial x}(\bar{A}(x)\bar{x} + \bar{B}(x, p_{in})) \right]_{x=\bar{x}, p_{in}=\bar{p}_{in}} (x - \bar{x}) \\
&\quad + \left[ \frac{\partial}{\partial p_{in}}(\bar{A}(x)\bar{x} + \bar{B}(x, p_{in})) \right]_{x=\bar{x}, p_{in}=\bar{p}_{in}} (p_{in} - \bar{p}_{in}) + \text{H.O.T.} \\
&= \bar{A}(\bar{x})\bar{x} + \bar{B}(\bar{x}, \bar{p}_{in}) + \left[ \bar{A}(x) + \frac{\partial \bar{A}}{\partial x} x + \frac{\partial \bar{B}}{\partial x} x = \bar{x}, p_{in}=\bar{p}_{in} \right] ((\bar{x} + \delta x) - \bar{x}) \\
&\quad + \left[ \frac{\partial \bar{B}}{\partial p_{in}} x = \bar{x}, p_{in}=\bar{p}_{in} \right] ((\bar{p}_{in} + \delta p_{in}) - \bar{p}_{in}) + \text{H.O.T.} \\
&\approx \dot{x} + \left( \bar{A}(\bar{x}) + \left[ \frac{\partial \bar{B}}{\partial x} x = \bar{x}, p_{in}=\bar{p}_{in} \right] \right) \delta x + \left[ \frac{\partial \bar{B}}{\partial p_{in}} x = \bar{x}, p_{in}=\bar{p}_{in} \right] \delta p_{in} \\
\delta \dot{x} &= \left( \bar{A}(\bar{x}) + \left[ \frac{\partial \bar{B}}{\partial x} x = \bar{x}, p_{in}=\bar{p}_{in} \right] \right) \delta x + \left[ \frac{\partial \bar{B}}{\partial p_{in}} x = \bar{x}, p_{in}=\bar{p}_{in} \right] \delta p_{in} \\
&= A\delta x + B\delta p_{in}
\end{align*} \]

Note that the partial derivative of \( \bar{A}(x) \) with respect to \( x \) involves derivatives of the Jacobians \( J_f, J_h \) and \( J_j \) with respect to \( u \), which are second-order with respect to \( u \) and can be absorbed into the higher order terms (H.O.T.), contributing to linearization error in the model when neglected. The matrix \( \bar{A}(x) \) is modified by the derivative of \( \bar{B}(x, p_{in}) \) with respect to \( x \), yielding \( A \); \( B \) is constructed from the partial derivative of \( \bar{B}(x, p_{in}) \) with respect to \( p_{in} \). In all cases, these terms are evaluated at the steady-state values of the state and input.

An output equation compatible with the desired state-space formulation for the system can be found through Taylor series approximation of \( y(x, p_{in}) = k(x) + l(p_{in}) \)
\[ \ddot{y} + \delta y = k(\ddot{x} + \delta x) + l(\ddot{p}_m + \delta p_m) \]
\[ = k(\ddot{x}) + l(\ddot{p}_m) + \left[ \frac{\partial k(x) + l(p_m)}{\partial x} \right]_{x=x, p_m=\ddot{p}_m} (x - \ddot{x}) \]
\[ + \left[ \frac{\partial k(x) + l(p_m)}{\partial p_m} \right]_{x=x, p_m=\ddot{p}_m} (p_m - \ddot{p}_m) + H.O.T \]
\[ \approx k(\ddot{x}) + l(\ddot{p}_m) + \left[ \frac{\partial k(x)}{\partial x} \right]_{x=\ddot{x}} ((\ddot{x} + \delta x) - \ddot{x}) \]
\[ + \left[ \frac{\partial l(p_m)}{\partial p_m} \right]_{p_m=\ddot{p}_m} ((\ddot{p}_m + \delta p_m) - \ddot{p}_m) \]
\[ \approx \ddot{y} + \left[ \frac{\partial k(x)}{\partial x} \right]_{x=\ddot{x}} \delta x + \left[ \frac{\partial l(p_m)}{\partial p_m} \right]_{p_m=\ddot{p}_m} \delta p_m \]
\[ \delta y = \left[ \frac{\partial k(x)}{\partial x} \right]_{x=\ddot{x}} \delta x + \left[ \frac{\partial l(p_m)}{\partial p_m} \right]_{p_m=\ddot{p}_m} \delta p_m \]
\[ = C\delta x + D\delta p_m \]

Like the state equation, H.O.T. related to higher derivatives and higher powers of perturbations \( \delta x \) and \( \delta p_m \), are negligible. The output matrix, \( C \), is the Jacobian of \( k(x) \) evaluated at \( \ddot{x} \) and, similarly, the feedthrough matrix, \( D \), is the Jacobian of \( l(p_m) \), evaluated at \( \ddot{p}_m \). The construction of each system matrix will be described subsequently.

**Constructing the state matrix.** The state matrix, \( A \), has the same block tridiagonal structure of \( \tilde{A}(x) \), in equation (2.40), and has modification to the block rows corresponding to \( k = 1 \) and \( k = N_z \) contributed from \( \partial \tilde{B}(x,p_m)/\partial x \). Three cases will be considered in defining these modifications: \( k = 1 \), \( k = N_z \) with supersonic flow, and \( k = N_z \) with subsonic flow.

For each case, \( B_{bc} \) at a specific grid point \((i^*, k^*)\) only has dependence on the flow conditions at that point. Then, the derivative with respect to \( x \) can be replaced by the derivative with respect to \( u_{i^*, k^*} \). This yields a matrix which contains terms only
in the block diagonal; that is, the submatrices in equations (2.34) to (2.39) will be modified, but those in equation (2.33) will not.

For the grid point on the isolator entrance plane, the modification to $\bar{A}(x)$ is given by:

$$
\left[ \frac{\partial}{\partial u} \left( - \frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) \right) \right]_{x=x} = \left[ - \frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} \frac{\partial B_{bc}^{in}}{\partial u} \right] - \frac{\partial}{\partial u} \left[ \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) \right]_{x=x} \approx - \frac{1}{\Delta T} I
$$

(2.49)

where H.O.T. (second derivative with respect to $u$) are neglected for simplicity. This same result is found when considering grid points in the exit plane with supersonic streamwise velocity.

When the streamwise velocity at the exit plane is subsonic, modification to $A(x)$ are contributed by $L_B(u, p_{in})$:

$$
\left[ \frac{\partial L_B}{\partial u} \right]_{x=x, u=u} = \left[ \frac{\partial}{\partial u} \left( \left[ \left( K_h^{-1} \right) N \right]^{-1} \left[ \begin{array}{c} 0 \\ -\frac{1}{\Delta T} B_{bc,su}^{out}(u, p_{in}) \end{array} \right] \right) \right]_{x=x, u=u} \\
= \left[ \frac{\partial}{\partial u} \left( \left( K_h^{-1} \right) N \right) \left[ \begin{array}{c} 0 \\ -\frac{1}{\Delta T} B_{bc,su}^{out}(u, p_{in}) \end{array} \right] \right]_{x=x, u=u} + \left[ \left( K_h^{-1} \right) N \right]^{-1} \left[ \begin{array}{c} 0 \\ -\frac{1}{\Delta T} B_{bc,su}^{out}(u, p_{in}) \end{array} \right]_{x=x, u=u} \approx \frac{1}{\Delta T} J_{mod}
$$

(2.50)
where $J_{mod}$ is given in equation (2.51). Here, the second-order derivatives with respect to $u$ are neglected, but would contribute to $B$ if included for more accuracy.

$$J_{mod} = \begin{bmatrix} -\frac{(\gamma-1)v\cdot v}{c^2} & \frac{(\gamma-1)u_r}{c^2} & \ldots & \frac{(\gamma-1)u_r}{c^2} \\ -\frac{(\gamma-1)v\cdot v}{c^2}u_r & -\frac{(\gamma-1)v\cdot v}{2c^2}(w-c) & \ldots & -\frac{(\gamma-1)v\cdot v}{c^2}(w-c) \\ -\frac{(\gamma-1)v\cdot v}{2c^2} \left(\frac{v\cdot v}{2} - wc + \frac{c^2}{\gamma-1}\right) & -\frac{(\gamma-1)u_r}{c^2} \left(\frac{v\cdot v}{2} - wc + \frac{c^2}{\gamma-1}\right) & \ldots & -\frac{(\gamma-1)u_r}{c^2} \left(\frac{v\cdot v}{2} - wc + \frac{c^2}{\gamma-1}\right) \\ (\gamma-1)u_r \frac{w}{c^2} & (\gamma-1)u_r \frac{w-c}{c^2} & \ldots & (\gamma-1)u_r \frac{w-c}{c^2} \\ (\gamma-1)u_r \frac{w-c}{c^2} & (\gamma-1)u_r \frac{w-c}{c^2} & \ldots & (\gamma-1)u_r \frac{w-c}{c^2} \end{bmatrix}$$ (2.51)

The modifications made to the submatrices in equations (2.34) to (2.39), for subsonic flow at the exit plane, are

$$A_{i,sub} = \tilde{A}_{i,sub} + \begin{bmatrix} -I & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & J_{mod} \end{bmatrix}$$ (2.52)

When the flow is supersonic at the exit plane, the first block row of $\tilde{A}_{i,sup} \ (\forall i)$ is replaced with

$$\begin{bmatrix} -I & 0 & \ldots & 0 \end{bmatrix}$$

and the final block row is replaced with

$$\begin{bmatrix} 0 & 0 & \ldots & 0 & -I \end{bmatrix}$$

to construct the submatrices $A_{i,sup}$. 45
The block tridiagonal state matrix, $A$, has the modified submatrices $A_i$ in the block diagonal and the submatrices $J_F^\pm$ in the block sub- and super-diagonals:

$$A = \begin{bmatrix}
A_1 & J_{F,1}^+ & \zeta & \zeta & \ldots & \zeta & J_{F,N-1}^- & A_{N-1} & J_{F,N}^- & A_N \\
J_{F,2}^+ & A_2 & J_{F,2}^- & \zeta & \ldots & \zeta & & & & \\
\zeta & J_{F,3}^+ & A_3 & J_{F,3}^- & \ldots & \zeta & & & & \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\zeta & \ldots & \ldots & J_{F,N-1}^+ & A_{N-1} & J_{F,N}^- & A_N & & & \\
\zeta & \ldots & \ldots & \ldots & \zeta & J_{F,N}^+ & A_N & & & 
\end{bmatrix} \quad (2.53)$$

Here, the flow velocity at $k = N_z$ dictates which submatrix will be chosen during model construction.

**Constructing the input matrix.** Contributions to the input matrix, $B$, can be found from the derivative of the last row of the submatrix in equation (2.41) or equation (2.42) with respect to $p_{in}$ and are dependent on the speed of the flow at the
exit plane. When the flow is subsonic at the exit plane, the non-zero block of $B$ is

$$B_{(i,Nz),sub} = \left[ \frac{\partial B}{\partial u} \right]_{x=\bar{x},p_{in}=\bar{p}_{in}}$$

$$= \left[ \frac{\partial}{\partial p_{in}} \left( \left[ (K_h^{-1})_N \right]^{-1} \begin{bmatrix} 0 \\ -\frac{1}{\Delta T} B_{bc,sub}^{out}(u, p_{in}) \end{bmatrix} \right) \right]_{x=\bar{x},p_{in}=\bar{p}_{in}}$$

$$= \left[ \left[ (K_h^{-1})_N \right]^{-1} \begin{bmatrix} 0 \\ -\frac{1}{\Delta T} \frac{\partial p_{in}}{\partial (p - p_{in})} \end{bmatrix} \right]_{x=\bar{x},p_{in}=\bar{p}_{in}}$$

$$= \frac{1}{\Delta T} \left[ \begin{bmatrix} \frac{1}{c^2} \\ \frac{u}{c} \\ \frac{w-c}{c^2} \\ \frac{1}{c^2} \left\{ \frac{v^2}{2} - wc + \frac{v^2}{\gamma-1} \right\} \end{bmatrix} \right]_{x=\bar{x}}$$

where the matrix being inverted has no dependence on the input because of the form of the boundary condition function.
In the case of supersonic exit flow, the submatrix $B_{(i,N_z),sup}$ is given by

$$B_{(i,N_z),sup} = \left[ \frac{\partial}{\partial p_{in}} \left( -\frac{1}{\Delta T} \left( \frac{\partial B_{bc,sup}^{out}}{\partial u} \right)^{-1} B_{bc,sup}^{out}(u, p_{in}) \right) \right]_{x=\bar{x}, p_{in}=\bar{p}_{in}}$$

$$= \left[ -\frac{1}{\Delta T} \left( \frac{\partial B_{bc,sup}^{out}}{\partial u} \right)^{-1} \frac{\partial B_{bc,sup}^{out}}{\partial p_{in}} \right]_{x=\bar{x}, p_{in}=\bar{p}_{in}}$$

$$= \frac{1}{\Delta T} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\gamma-1}{2} v \cdot v & -(\gamma-1)u & -\gamma(u-1)w & (\gamma-1) \\ 0 & 0 & 0 & 1 \end{bmatrix}_{x=\bar{x}}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (2.55)$$

$$= \frac{1}{\Delta T} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{w(\gamma-1)} \\ 0 \end{bmatrix}_{x=\bar{x}}$$

where, again, $\partial B_{bc,sup}^{out}/\partial u$ has no dependence on $p_{in}$ due to the definition of the boundary condition function.

The $B$ matrix is built from equations (2.54) and (2.55) where flow velocity information from the steady-state CFD data determine which submatrix will be chosen in construction of equation (2.56). Note that there will be $N_z - 1$ block rows of zeros.
between each $B_{i,N_z}$:

\[
B = \begin{bmatrix}
Z \\
\vdots \\
Z \\
B_{1,N_z} \\
Z \\
\vdots \\
Z \\
B_{2,N_z} \\
Z \\
\vdots \\
\vdots \\
B_{N_r,N_z}
\end{bmatrix}
\tag{2.56}
\]

**Constructing the output matrix.** Consider a particular row of $k(x)$ (that is, $k_{i*,k*}(u_{i*,k*})$, which defines the output for an index pair $(i*, k*)$. Since this function only depends on $u_{i*,k*}$, its Jacobian will be the non-zero block element in the row of $C$ corresponding to this output. Denoting this Jacobian by $C_{i*,k*}$, it can be evaluated by taking the partial derivative of $k_{i*,k*}(u_{i*,k*})$ with respect to $u_{i*,k*}$.

Upstream of the exit plane (grid points with, $k = [1 \ 1 \ k_2 \ k_3]$), pressure and temperature can be expressed in terms of the flow variables, and $C_{i,k}$ is

\[
C_{i,k} = \begin{bmatrix}
-\frac{u_r}{\rho} & \frac{1}{\rho} & 0 & 0 \\
-\frac{w}{\rho} & 0 & \frac{1}{\rho} & 0 \\
(\gamma - 1)\frac{v \cdot v}{2} & -(\gamma - 1)u_r & -\frac{\gamma - 1}{\rho} & (\gamma - 1) \\
\frac{\gamma - 1}{R} \left[ -\frac{E}{\rho} + \frac{v \cdot v}{\rho} \right] & -\frac{\gamma - 1}{R} \frac{w}{\rho} & -\frac{\gamma - 1}{R} \frac{w}{\rho} & \frac{\gamma - 1}{R} \frac{w}{\rho}
\end{bmatrix}
\tag{2.57}
\]
At the exit plane \((k = N_z)\), the Jacobian of \(k'(u_{i,N_z})\) is \(C_{i,N_z}\), which is equation \((2.57)\) with the third row (pressure) equal to 0 since pressure is an input at the exit plane:

\[
C_{i,N_z} = \begin{bmatrix}
-\frac{u_r}{\rho} & \frac{1}{\rho} & 0 & 0 \\
-\frac{w}{\rho} & 0 & \frac{1}{\rho} & 0 \\
0 & 0 & 0 & 0 \\
\gamma - 1 \left[ -\frac{E}{\rho} + \frac{v \cdot v}{\rho} \right] & -\frac{\gamma - 1}{\gamma - 1} \frac{u_r}{\rho} & -\frac{\gamma - 1}{\gamma - 1} \frac{w}{\rho} & \gamma - 1 \\
\end{bmatrix}
\]

(2.58)

The contributions of non-feedthrough terms, equations \((2.57)\) and \((2.58)\), can be combined to form the output matrix, \(C\), which has \(N_r\) submatrices \(C_i\) with dimensions 20-by-4:

\[
C_i = \begin{bmatrix}
C_{i,1} & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
0 & \ldots & 0 & C_{i,k_2} & 0 & \ldots & 0 & 0 & \ldots & 0 \\
0 & \ldots & 0 & 0 & \ldots & 0 & C_{i,k_3} & 0 & \ldots & 0 \\
0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & C_{i,k_4} \\
0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & C_{i,N_z}
\end{bmatrix}
\]

where \(C_{i,k_i}\) is in the \(k_i^{th}\) block column of \(C_i\). Arranging these matrices in a block-diagonal structure forms the 20\(N_r\)-by-4\(N_r\)\(N_z\) output matrix

\[
C = \begin{bmatrix}
C_1 & \zeta & \zeta & \ldots & \zeta & \zeta \\
\zeta & C_2 & \zeta & \ldots & \zeta & \zeta \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\zeta & \ldots & \zeta & \ldots & \zeta & C_{N_r}
\end{bmatrix}
\]

(2.59)

**Building the feedthrough matrix.** For grid points located upstream of the isolator exit plane \((k = [1 \ k_1 \ k_2 \ k_3])\), the 4-by-1 block element of \(D\) is the zero-vector. The non-zero block elements of \(D\) corresponds to those point in the isolator exit plane.
(k = N_z); for the cell identified by index pair (i, N_z), the 4-by-1 matrix D_{i,N_z} is found as

\[ D_{i,N_z} = \frac{\partial l(p_{in})}{\partial p_{in}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \]  \hspace{1cm} (2.60) 

where the system input, p_{in}, is scalar because of the assumption that the imposed backpressure (by the combustor) is constant across the entire exit plane. The intermediate feedthrough matrix D_i, with dimension 20-by-1, is

\[ D_i = \begin{bmatrix} Z \\ Z \\ Z \\ Z \\ D_{i,N_z} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix} \]

where Z is the 4-by-1 zero matrix. These vectors can be augmented to form the 20N_r-by-1 feedthrough matrix

\[ D = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{N_r} \end{bmatrix} = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & \ldots & 0 & 1 & 0 \end{bmatrix}^T \]  \hspace{1cm} (2.61)
Evaluating the SPM. The SPM, which has an LTI form, can be expressed given the matrices in equations (2.53), (2.56), (2.59) and (2.61):

\[
\begin{align*}
\delta \dot{x} &= A\delta x + B\delta p_m \\
\delta y &= C\delta x + D\delta p_m
\end{align*}
\tag{2.62}
\]

where \( \delta x \in \mathbb{R}^{4N_rN_z} \), \( \delta u \in \mathbb{R}^1 \), \( \delta y \in \mathbb{R}^{20N_r} \) and the system matrices take on the appropriate dimensions.

A numeric form of the SPM is found by evaluating each matrix \( A, B, C, \) and \( D \) at a steady-state CFD solution provided by AFRL. This data corresponds to the steady-state solution found for a backpressure \( p_{b,A} = 218.577 \) kPa imposed on the isolator exit plane. It is not practical to provide the numerical details of the large system matrices, but the structures can be seen in Figures 2.3 and 2.4. The block tridiagonal structure of \( A \) and the block diagonal structure of \( C \) are readily observed in these figures, and it can be noted that \( B \) and \( D \) are sparse matrices, with non-zero elements only where the input directly appears in the governing equations.

Model Validation

After the SPM has been constructed, the accuracy of the model in capturing the flow dynamics of the system must be evaluated. This will help determine if the SPM is adequate for use in the unstart control problem. Before simulation, the model was studied for stability to ensure it had been correctly constructed. Unexpected instability was encountered, found to be related to the index scheme along the radial axis. Upon resolving these problems, several simulations were run to provide insight into the usefulness of the SPM.

Building a stable model. The isolator model was constructed for a grid defined by cylindrical coordinates \( (r, z) \). The first index scheme assigned \( i = 1 \) at the isolator
Figure 2.3: Visualization of the structures of the state and input matrices $A$ and $B$ for the SPM. The tridiagonal structure of $A$ is clear from these plots.

Figure 2.4: Visualization of the structures of the output and feedthrough matrices $C$ and $D$ for the SPM. The diagonal structure of $C$ is clear from these plots.
wall, as this was the order of the data exported from CFD simulation. The eigenvalues of $A$ for this model were calculated using resources at the Ohio Supercomputer Center (OSC) and found to be in both the RHP and LHP (Figure 2.5), indicating an unstable model. As a test, a model was constructed assuming a rectangular coordinate system ($x, z$) and the eigenvalues of the state matrix of this model were found to be in the LHP only, as can be seen in Figure 2.5. The maximum eigenvalue of $\lambda_{\text{max}} = -1$, suggesting that the model was stable in this coordinate system.

The isolator model was expected to be stable independent of the coordinate system for which it is constructed, so the CFD data and grid were studied more closely to find the reason for the instability. Since the isolator geometry strongly suggested defining the model in cylindrical coordinates, it was of interest to find why this formulation yielded an unstable model. It was found that the definition of the isolator grid for CFD simulation was responsible for the model’s instability; redefinition of the index scheme was found to produce a stable model in cylindrical coordinates.

Ordering the mesh points. In constructing the intermediate isolator model, an index pair $(i, k)$ was assigned to each point on the grid in Figure 2.6, where $i \in [1, N_r]$ and $k \in [1, N_z]$, as described in Section 2.1. The original index scheme (‘index scheme 1’), illustrated in Figure 2.2 in both rectangular and cylindrical coordinate systems, was assigned by TecPlot360, the software used to export CFD simulation results to a text file for use in MATLAB®. Here, it can be seen that order in which the index $k$ is defined is the same in both coordinate systems ($k = 1$ at the entrance), while the order of $i$ is not. When considering rectangular coordinates, $i$ increases as $x$ increases, but in cylindrical coordinates $i$ decreases as the radial coordinate, $r$, increases. This causes the finite-difference $r_{i+1} - r_i$ to be negative in the cylindrical coordinate system when it is expected to be positive.
Figure 2.5: Eigenvalues of $A$ for the SPM constructed in rectangular and cylindrical coordinates, constructed using index scheme 1, where radial indexing begins at the wall. For the stable model (rectangular coordinates), the largest eigenvalue is $\lambda_{\text{max}} = -1$.

Figure 2.6: Schematics showing the mesh and indexing scheme used by TecPlot360, in relation to both the $(x,z)$ and $(r,z)$ planes, on the left and right, respectively.
Figure 2.7: Example indexing schemes for the radial direction, showing how they relate to the \(x\) and \(r\) coordinates in the \(\theta = \pi\) case. Index scheme 1, on the left, is given by TecPlot360, where \(i\) increases with \(x\), while index scheme 2, on the right, is assigned so that \(i\) increases with \(r\).

Since the model in rectangular coordinates was stable, it was of interest to ‘reverse’ the order of \(i\) along the \(r\) axis, so that \(i = 1\) at \(r = \epsilon\) and \(i = N_r\) at \(r = R(z)\) (‘index scheme 2’). For this redefined scheme, \(i\) increased as \(r\) increased, and decreased as \(x\) increased. The plots in Figure 2.7 show how these indexing schemes compare schematically on both the \(x\) and \(r\) axes.

Since the order of the indices is reversed for index scheme 2, \(x_{i+1} - x_i\) is negative while \(r_{i+1} - r_i\) is positive. It is expected that the finite-differences calculated along the coordinate axis should be positive, so the index scheme should enforce this for a particular grid. The isolator grid in Figure 2.2 is defined for the azimuthal plane at \(\theta = \pi\), so different indexing schemes are defined depending on the coordinate system of interest: index scheme 1 for rectangular coordinates and index scheme 2 for cylindrical coordinates.

In order to verify that these changes produced a stable model in cylindrical coordinates, the CFD data was re-ordered to follow index scheme 2, and models were constructed in both rectangular and cylindrical coordinates. The OSC was used to evaluate the eigenvalues of the state matrices of each of these models (Figure 2.8) and, as expected, the model constructed in cylindrical coordinates was stable (\(\lambda_{\text{max}} = -1\)) while that constructed in rectangular coordinates was not.
Figure 2.8: Eigenvalues of $A$ for the SPM constructed in rectangular and cylindrical coordinates using index scheme 2, where radial indexing begins at the center-line. For the stable model (cylindrical coordinates), the largest eigenvalue is $\lambda_{\text{max}} = -1$.

Table 2.2: Run times for each of the SPM simulations, based on the $\text{tic}$ and $\text{toc}$ timing functions available in MATLAB®.

<table>
<thead>
<tr>
<th></th>
<th>Rectangular</th>
<th>Cylindrical</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta u_1$</td>
<td>34604.2512 sec (9h 36m 44.25s)</td>
<td>34631.1592 sec (9h 37m 11.16s)</td>
</tr>
<tr>
<td>$\delta u_2$</td>
<td>34604.1412 sec (9h 36m 44.14s)</td>
<td>34569.9734 sec (9h 36m 9.97s)</td>
</tr>
</tbody>
</table>
Simulation Results. The two stable SPMs were constructed, the model in rectangular coordinates constructed with index scheme 1 (‘rectangular formulation’) and the model in cylindrical coordinates constructed with index scheme 2 (‘cylindrical formulation’), and simulated under two backpressure perturbations: $\delta u_1 = p_{bC} - p_{bA}$ (+3.54%) and $\delta u_2 = p_{bD} - p_{bA}$ (+7.87%). AFRL provided data from transient CFD simulations modeling the response of the system as it transitions from the steady-state condition due to backpressure $p_{bA} = 218.577$ kPa to a final steady-state condition, due to backpressures $p_{bC} = 226.313$ kPa or $p_{bD} = 235.78$ kPa. This data was collected for 50 ms, at 1 ms intervals, the same time intervals for which data was collected during simulation of the SPM.

The simulation was run using the `ode45` solver in 64-bit MATLAB®, running on a Dell Precision T1500 workstation with 64-bit Windows 7, an Intel Core i5 2.80 GHz processor, and 16 GB RAM. Despite the high-performance hardware, it took between 9.5 and 10 hours to complete each simulation due to the large size of the model; run times were measured using built-in timer functions in MATLAB® and can be seen in Table 2.2. These long simulation run-times suggest that the SPM would need to be reduced for control-design applications, because it is not practical to tune a controller for such a large model.

Results and Analysis. The output data collected for each model contains transient results for the velocity components, pressure and temperature in planes at the streamwise locations given in Table 2.3. Since the intervals in the isolator grid are non-uniform, these locations were determined analytically for the particular grid in Figure 2.2.

Because the output vector has a dimension $q = 980$, it is necessary to focus on a reduced set of outputs; in this case, outputs at the center-line, (radial) midpoint, and wall of the isolator. Since each output had a different order or magnitude (e.g.,
Table 2.3: Streamwise locations of output sensors, with corresponding index \( k \).

<table>
<thead>
<tr>
<th>( k )</th>
<th>1</th>
<th>37</th>
<th>67</th>
<th>99</th>
<th>149</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z ) (m)</td>
<td>0.1829</td>
<td>0.4060</td>
<td>0.6330</td>
<td>0.8636</td>
<td>1.0897</td>
</tr>
</tbody>
</table>

radial velocity was on the order of 10 while pressure was on the order of 10000), it was informative to consider each as a percent error calculated in relation to the initial steady-state and expected transient values (from CFD):

\[
\%Err = \frac{(\cdot)_{sim} - ((\cdot)_{CFD} - (\cdot)_{SS,A})}{(\cdot)_{SS,A}} (100)
\]

\[
= \frac{(\cdot)_{sim} - \delta(\cdot)_{CFD}}{(\cdot)_{SS,A}} (100)
\]

where (\cdot) represents the output of interest (that is, \( u_r, w, p \) or \( T \)). Using percent errors allows for the results to be shown more compactly and analyzed more easily.

Results for input \( \delta u_1 = p_{bC} - p_{bA} \). Figures 2.9 to 2.11 show the output errors, calculated using equation (2.63), for the model simulated with the input \( \delta u_1 = p_{bC} - p_{bA} (+3.54\% \text{ perturbation}) \). A summary of steady-state errors estimated from the plots can be found in Table 2.4.

Along the center-line (Figure 2.9), it is interesting to note the behavior of the signals at the two most-downstream sensor locations. As indicated in Figure 2.15, these sensors are downstream of the original steady-state shock location. At the center-line, each output of the rectangular formulation of the model diverges at \( z = 0.8636 \) m, and both formulations have outputs that diverge at the exit plane.

Upstream of the original shock, the outputs establish a steady-state early in the simulation. It should be noted that the large errors in the radial velocity (\( u_r \)) are related to the small magnitude of the expected output (around 5 m/s), since the model output vanished except around the shock location. Upstream of the shock,
Figure 2.9: Percent-errors, calculated relative to steady-state condition $\bar{x}$, of the outputs at the isolator center-line from simulation with input $\delta u_1$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.

The relative errors of each output, except pressure, increased the further downstream they are measured.

At the center-line, the profiles of each flow variable ($\rho$, $u_r$, $w$, and $e$) were different from those at other radial locations due to spurious behavior in the CFD data. Because of this anomalous behavior, it was helpful to look at the output in two other radial planes: the radial midpoint and the wall. These results are shown in Figures 2.10 and 2.11, respectively.

In both planes, outputs of the rectangular formulation of the SPM diverges at the two most-downstream locations, like at the center-line. For the cylindrical formulation, only the radial velocity diverged at the exit plane. It can also be observed that these errors are smaller as they are measured further from the center-line.
Figure 2.10: Percent-errors, calculated relative to steady-state condition $\bar{x}$, of the outputs at the radial midpoint from simulation with input $\delta u_1$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.

Figure 2.11: Percent-errors, calculated relative to steady-state condition $\bar{x}$, of the outputs at the isolator wall from simulation with input $\delta u_1$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.
Results for input $\delta u_2 = p_{bD} - p_{bA}$. The output errors in Figures 2.12 to 2.14 are calculated using equation (2.63) with outputs from simulation of the SPM with the input $\delta u_2 = p_{bD} - p_{bA}$ (+7.87% perturbation). The steady-state values of these errors are summarized in Table 2.4.

Along the center-line (Figure 2.12), the output errors are the two most-downstream sensor locations are similar to those calculated from simulation with the input $\delta u_1$ (Figure 2.9): each output diverges in both locations for the rectangular formulation, and only at the exit plane for the cylindrical formulation.

Upstream of the original shock, each output establishes a steady-state. For these simulations, it should be noted that the third sensor is located downstream of the expected final shock location, which is hypothesized to contribute to the large output errors at this location. These large errors were not observed for simulation with $\delta u_1$ since the shock is not expected to propagate past any of the sensors.

Like simulation with $\delta u_1$, the outputs from the simulation with $\delta u_2$ were also studied at the radial midpoint (Figure 2.13) and at the wall (Figure 2.14). In both planes, the output error for the simulation of the rectangular formulation of the SPM diverged at the two most-downstream sensor locations. Only the errors of the radial velocity diverged at the exit plane for the cylindrical model formulation. This behavior was similar to that observed from simulation with input $\delta u_1$. These are similar observations to those made from results from simulation with $\delta u_1$.

The steady-state estimates in Table 2.4 were determined from the plots in Figures 2.9 to 2.14. Errors at the first two sensor locations, for both simulations, are expected to be similar as the final steady-state shock locations are downstream of both sensors and the flow fields do not change much from the initial conditions in these regions (as can be seen in Figure 2.15). It should also be noted that, at the exit plane, the pressure error is much smaller than the errors of the other outputs because
Figure 2.12: Percent-errors, calculated relative to steady-state condition $\bar{x}$, of the outputs at the isolator center-line from simulation with input $\delta u_2$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.

Figure 2.13: Percent-errors, calculated relative to steady-state condition $\bar{x}$, of the outputs at the radial midpoint from simulation with input $\delta u_2$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.
Figure 2.14: Percent-errors, calculated relative to steady-state condition $\bar{\vec{x}}$, of the outputs at the isolator wall from simulation with input $\delta u_2$. Each subplot shows the four outputs, for rectangular and cylindrical model formulations, at the streamwise location indicated by the vertical axis label.

it is calculated directly from the input, a constant value within numerical error of the CFD results.
Table 2.4: Steady-state percent-errors of simulation outputs, estimated from the plots in Figures 2.9 to 2.14. The streamwise and radial locations of each output are indicated along with the input. When indicated ((c) or (r)), estimates only apply to that formulation, as they otherwise diverge. (Empty cells indicate diverging behavior for both model formulations.)

<table>
<thead>
<tr>
<th>Input</th>
<th>Radial Location</th>
<th>Output</th>
<th>$z = 0.1829$ m</th>
<th>$z = 0.4060$ m</th>
<th>$z = 0.6330$ m</th>
<th>$z = 0.8636$ m</th>
<th>$z = 1.0897$ m</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{\text{C}}$</td>
<td>center-line</td>
<td>$u$</td>
<td>3.4%</td>
<td>-102%</td>
<td>15%</td>
<td>-110% (c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w$</td>
<td>0.26%</td>
<td>-20.9%</td>
<td>-28.5%</td>
<td>16.5% (c)</td>
<td>-194% (r)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p$</td>
<td>$10^{-4}$%</td>
<td>-0.02%</td>
<td>-0.011%</td>
<td>-29.6% (c)</td>
<td>-2 $\cdot 10^{-3}$%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T$</td>
<td>0.1%</td>
<td>37.5%</td>
<td>41.35%</td>
<td>-8.35% (c)</td>
<td>36.1% (r)</td>
</tr>
<tr>
<td>midpoint</td>
<td>$u$</td>
<td>$1.5 \cdot 10^{-3}$%</td>
<td>-34.5%</td>
<td>0.28%</td>
<td>-15% (c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$w$</td>
<td>$10^{-5}$%</td>
<td>-0.5%</td>
<td>$10^{-3}$%</td>
<td>26% (c)</td>
<td>-0.3% (c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p$</td>
<td>$-3.5 \cdot 10^{-5}$%</td>
<td>$2.2 \cdot 10^{-3}$%</td>
<td>$-2.4 \cdot 10^{-3}$%</td>
<td>-30% (c)</td>
<td>$-5 \cdot 10^{-3}$%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T$</td>
<td>$-10^{-9}$%</td>
<td>$1.4 \cdot 10^{-3}$%</td>
<td>$-10^{-3}$%</td>
<td>-12% (c)</td>
<td>-1.5% (c)</td>
<td></td>
</tr>
<tr>
<td>wall</td>
<td>$u$</td>
<td>$O(10^{-3})$%</td>
<td>$O(10^{-9})$%</td>
<td>0.5 $\cdot 10^{-4}$%</td>
<td>-28% (c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$w$</td>
<td>$O(10^{-3})$%</td>
<td>$O(10^{-9})$%</td>
<td>$-1.5 \cdot 10^{-3}$%</td>
<td>25.19% (c)</td>
<td>-2% (c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-9})$%</td>
<td>$2 \cdot 10^{-3}$%</td>
<td>-30.5% (c)</td>
<td>$-2 \cdot 10^{-3}$%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$T$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-9})$%</td>
<td>$-10^{-3}$%</td>
<td>-11.75% (c)</td>
<td>-1% (c)</td>
<td></td>
</tr>
<tr>
<td>$p_{\text{D}}$</td>
<td>center-line</td>
<td>$u$</td>
<td>3.4%</td>
<td>-102%</td>
<td>-200%</td>
<td>-105% (c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w$</td>
<td>0.26%</td>
<td>-20.9%</td>
<td>50%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p$</td>
<td>$-0.5 \cdot 10^{-7}$%</td>
<td>-0.01%</td>
<td>-380%</td>
<td>-29.5% (c)</td>
<td>$-4 \cdot 10^{-3}$%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T$</td>
<td>0.095%</td>
<td>37.3%</td>
<td>-0.1%</td>
<td>-10% (c)</td>
<td></td>
</tr>
<tr>
<td>midpoint</td>
<td>$u$</td>
<td>$1.5 \cdot 10^{-3}$%</td>
<td>35%</td>
<td>88%</td>
<td>-27% (c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w$</td>
<td>$O(10^{-5})$%</td>
<td>$-1.5 \cdot 10^{-3}$%</td>
<td>63.2%</td>
<td>22% (c)</td>
<td>-3.7% (c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p$</td>
<td>$O(10^{-5})$%</td>
<td>$2.5 \cdot 10^{-3}$%</td>
<td>-364.75%</td>
<td>-30% (c)</td>
<td>$-7 \cdot 10^{-3}$%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T$</td>
<td>$O(10^{-5})$%</td>
<td>$2 \cdot 10^{-4}$%</td>
<td>$-68.8$%</td>
<td>-14% (c)</td>
<td>-3.2% (c)</td>
</tr>
<tr>
<td>wall</td>
<td>$u$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-2})$%</td>
<td>64.5%</td>
<td>-32.3% (c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-2})$%</td>
<td>64.25%</td>
<td>24.25% (c)</td>
<td>-0.35% (c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$p$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-2})$%</td>
<td>-362%</td>
<td>-30.35% (c)</td>
<td>$= 3.5 \cdot 10^{-3}$%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T$</td>
<td>$O(10^{-5})$%</td>
<td>$O(10^{-2})$%</td>
<td>$-68.2$%</td>
<td>$-14.25$% (c)</td>
<td>-3% (c)</td>
</tr>
</tbody>
</table>
The most important conclusion that can be drawn from these simulation results is that shock motion is not captured by the SPM. This can be seen from the large errors at sensor locations between the initial and final shocks, most notably when the input is $\delta u_2$. Figure 2.15, which shows radially-averaged percent perturbations of pressure, Mach number, and temperature at steady-state, calculated from CFD data, for both perturbed inputs, provides further support for this conclusion. Note that these profiles are not like the percent errors calculated for the outputs, but instead show how the flow-field changes due to a change in backpressure.

In the figure, the shaded boxes indicate where each shock wave is located in steady-state, with the shock farthest downstream corresponding to the initial shock location (when backpressure is $p_{bA}$), and the other boxes show the final steady-state shock locations due to increases in backpressure to $p_{bD}$ and $p_{bC}$ (left to right). The output is measured at locations indicated by dashed lines; the first location downstream of the entrance plane is upstream of all three shocks, while the second is downstream of the shock that forms due to $p_{bD}$ is imposed on the isolator. The third location is near the original shock, which may explain why outputs measured here behave differently from more-upstream outputs.

These profiles shed some light on changes in the flow field under input perturbations. In regions upstream and downstream of all shocks, the flow fields do not significantly change. The percent perturbations of each flow variable (pressure, Mach number, and temperature) in the upstream region are on the order of $10^{-4}$%, as expected since the downstream input perturbation does not change the flow field upstream of the shock [28, 29]. In the downstream region, perturbations in these three flow variables are under 10% for both inputs (4%, 3%, and 0.3%, respectively, for backpressure $p_{bC}$, and 8%, 6%, and 0.6% for backpressure $p_{bD}$). Perturbations in the exit plane due to $p_{bD}$ are expected to be larger since $p_{bD} > p_{bC}$.
Profiles of pressure, Mach number and temperature in the region of the isolator between the initial and final shock locations provide the most useful information for understanding the simulation results. In this region, the perturbations are on the ranges $350 - 380\%$, $71.5 - 73\%$, and $69 - 72\%$, respectively, for backpressure $p_{bC}$, and $340 - 400\%$, $70.5 - 73.5\%$, and $66 - 72.5\%$, respectively, for backpressure $p_{bD}$. When the input is $p_{bD}$, this region of the isolator contains the third sensor ($z = 0.6330$ m), where the percent error of the simulation results was found to be the greatest. This large output error suggests that the SPM does not capture any change in the flow field. If the output were redefined to include a sensor downstream of the shock that forms for $p_{bC}$ it would be expected that the output error at this location would likewise be larger. The inability of the SPM to capture changes in the flow field
suggests that the linearized model is unable to capture any shock motion due to an input perturbation.

**Conclusions**

During the model validation process, the SPM constructed following the methodology presented in Refs. 28, 29, was found to be inadequate for use in the unstart control problem because it does not capture shock motion. Because the SPM is a linearized model, it is only accurate in capturing small changes from the steady-state about which it is constructed and cannot capture the large perturbations in some regions of the flow field that occur due to small input perturbations. For designing a controller to prevent unstart, which occurs when the shock is dislodged from the isolator, it is not necessary that the model accurately captures the entire flow field, but it should *at least* capture shock motion.

Although the SPM may be used to build a surrogate model of the shock motion, several sets of CFD solutions would be required to construct this model since the output errors cannot be determined from simulation of the SPM alone. An alternate option may be to construct the isolator model from a ‘set’ of SPMs evaluated for various backpressures, similar to the trajectory piecewise-linear model approach by Gratton and Willcox in Ref. [37]. This would be most appropriate if the full flow-field were of interest. It may instead be necessary to consider a more complex PBM, such as the nonlinear model in equation (2.29), or to consider a different modeling method altogether, like the proper-orthogonal decomposition (POD) and a projection method implemented, for example, in Refs. 37–39. To maintain consistency with the SPM, a nonlinear model is constructed from equation (2.29) in the next section.
2.2.9 Nonlinear model

A more complex PBM was constructed through expansion of equation (2.29) to a form affine with respect to the input, \( \dot{x} = A(x)x + B(x)p_{in} \). This is possible through application of Euler’s theorem of homogeneous functions to \( \tilde{B}(x, p_{in}) \), since the boundary condition functions are homogeneous functions of degree one. This expansion produces terms that contribute to the construction of \( A(x) \) and also terms from which \( B(x) \) can be constructed.

It was found that the state and input matrices constructed from the expansion of \( \tilde{B}(x, p_{in}) \) had the same structure as those constructed in Section 2.2.8. Specifically, \( A(x) \) in equation (A.5) and \( B(x) \) in equation (A.8) are the state-dependent forms of the LTI matrices \( A \) (equation (2.53)) and \( B \) (equation (2.56)) constructed for the SPM. The details of the construction of \( A(x) \) and \( B(x) \) are similar to those presented for construction of \( A \) and \( B \), and can be found in Appendix A.

The nonlinear PBM of the isolator is completed through definition of an output function for the model. Unlike the SPM, only one output variable is measured at each grid point, simplifying the structures of the matrices \( C(x) \) and \( D \) in comparison to those presented in Section 2.2.8.

Constructing the Nonlinear Output Function

For the nonlinear model, the output function should be expressed in a form that is affine with respect to the input, \( y = C(x)x + Dp_{in} \). Instead of measuring four variables at each grid point of interest, the only flow variable of interest is the pressure, as it is possible to determine the shockwave location from a pressure profile along the length of the isolator. To maintain generality, the locations at which the output is measured are not specified and the general structure of \( C(x) \) is instead discussed.
In practice, pressure along the wall is the simplest measurement to make in experiments and therefore would be of most interest as a model output for comparison with experimental measurements.

For a given grid point, pressure \( p = (\gamma - 1)\rho(e - \frac{v \cdot v}{2}) \) is a homogeneous function of degree one and can therefore be expressed as \( p = (\partial p/\partial u)u \). For a given grid point, with indices \((i^*, k^*)\), upstream of the exit plane, the block element of \( C(x) \) is the Jacobian \((\partial p/\partial u)|_{i^*,k^*} \):

\[
p_{i,k} = \frac{\partial p}{\partial u} \bigg|_{i,k} = \left[ \frac{\gamma - 1}{2} v \cdot v - (\gamma - 1)u_r - (\gamma - 1)w \right. - (\gamma - 1)w \left. \right]_{i,k} u_{i,k} \tag{2.64}
\]

\[
= C_{i,k}(u)u_{i,k}
\]

The output matrix \( C(x) \) has dimensions \( q \)-by-\( n \), where \( q \) is the number of locations at which pressure is of interest. The output vector is formed by stacking the output signals with an index order reflective of equation (2.16):

\[
y = \begin{bmatrix}
p_{i_1,k_1} \\
p_{i_1,k_2} \\
\vdots \\
p_{i_1,k_{end}} \\
p_{i_2,k_1} \\
\vdots \\
p_{i_{end},k_{end}}
\end{bmatrix} \tag{2.65}
\]

where the subscripts on \( i \) and \( k \) are the ‘index of the output index,’ where the indices of the locations at which pressure is measured are arranged in increasing order \((i \text{ toward the wall}, k \text{ downstream})\).
When \( k \neq N_z \), the output for point \((i, k)\) (call it output \( q^* \)), given by equation (2.64), is calculated from the flow vector \( \mathbf{u}_{i,k} \) containing states \( x_{4((i-1)N_z+k-1)+1} \) to \( x_{4((i-1)N_z+k)} \). The subscripts on \( x \) that define \( \mathbf{u}_{i,k} \) also define the column range where \( C_{i,k}(\mathbf{u}) \) is inserted in row \( q^* \) of \( C(x) \). Because of the structure of \( y \), \( C(x) \) is a block diagonal matrix, where the size of each block depends on the number of points at which pressure measurements are of interest.

At the isolator exit plane \((k = N_z)\), the output is \( p_{in} \), a direct feedthrough of the the input. This means that the row of \( C(x) \) associated with this output is zero and the feed-through matrix, \( D \), has a 1 in the same row. Consequently, \( D \) is not a state-dependent matrix, but instead contains a pattern of 1’s and 0’s that depends on the locations at which pressure measurements are made.

Through the definitions of \( C(x) \) and \( D \) given in this section, along with the matrices \( A(x) \) and \( B(x) \) in equations (A.5) and (A.8), the nonlinear system can be expressed in affine form:

\[
\dot{x} = A(x)x + B(x)p_{in} \\
y = C(x)x + Dp_{in}
\]

(2.66)

where \( x \in \mathbb{R}^{4N_rN_z} \), \( u \in \mathbb{R} \), \( y \in \mathbb{R}^q \) and each matrix has compatible dimensions.

**Damping Modification**

Simulation of the nonlinear model revealed that, instead of capturing shock motion, the pressure profiles showed a build-up of pressure downstream of the shock, without affecting upstream flow conditions. A similar artifact is observed in CFD simulations with the Euler equations due to the absence of viscosity in the model, and can be addressed by including one of several forms of artificial dissipation (also referred to as artificial viscosity or damping), such as those described by Hirsch in Ref. 34, in the model.
Many of these dissipation schemes involve inclusion of higher-order state derivatives in the model being simulated by the CFD software. Weighting most heavily the simplicity of the scheme and its shock-capturing ability, the two-part scheme described by Swanson and Turkel [40] and Pulliam [41], which expands the work of Jameson, et al. [42], was adapted for modification of the nonlinear, affine model in equation (2.66).

The damping scheme has two terms: a second-order term \textit{added} to the RHS of equation (2.66), and a fourth-order term \textit{subtracted} from the RHS of the equation. These terms are given by Swanson and Turkel in discretized form [40], and it is not straight-forward to express them in continuous spatial derivatives because the grid is non-uniform. However, considering a uniform grid, it is possible to see the general nature of each term: the second-order term is a second derivative weighted by the square of the grid spacing, while the fourth-order term is a fourth-derivative weighted by the fourth power of the grid spacing.

Using the notation of in Ref. 40, where the backwards-difference operator is $\nabla$ and the forward-difference is $\Delta$, the damping term for grid point $(i, k)$ is

$$
(D_r^2 + D_z^2 - D_r^4 - D_z^4)\mathbf{u} = \{\nabla_r [\lambda_{i+0.5,k} \epsilon_{i+0.5,k}^2 \Delta_r] + \nabla_z [\lambda_{i,k+0.5} \epsilon_{i,k+0.5}^2 \Delta_z]
- \nabla_r [\lambda_{i+0.5,k} \epsilon_{i+0.5,k}^4 \Delta_r \nabla_r \Delta_r]
- \nabla_z [\lambda_{i,k+0.5} \epsilon_{i,k+0.5}^4 \Delta_z \nabla_z \Delta_z] \} \mathbf{u}_{i,k}
$$

(2.67)
with scaling factors

\[
\begin{align*}
\lambda_{i+0.5,k} &= \frac{1}{2} [\lambda_{r,(i+1,k)} + \lambda_{r,(i,k)} + \lambda_{z,(i+1,k)} + \lambda_{z,(i,k)}] \\
\lambda_{i,k+0.5} &= \frac{1}{2} [\lambda_{r,(i,k+1)} + \lambda_{r,(i,k)} + \lambda_{z,(i,k+1)} + \lambda_{z,(i,k)}] \\
\epsilon^2_{i+0.5,k} &= \kappa^2 \max \{\nu_{i,k}, \nu_{i+1,k}\} \\
\epsilon^2_{i,k+0.5} &= \kappa^2 \max \{\nu_{i,k}, \nu_{i,k+1}\} \\
\epsilon^4_{i+0.5,k} &= \max \{0, \kappa^4 - \epsilon^2_{i+0.5,k}\} \\
\epsilon^4_{i,k+0.5} &= \max \{0, \kappa^4 - \epsilon^2_{i,k+0.5}\}
\end{align*}
\]

(2.68)

Let ‘*’ represent a forward or backwards shift of \(1/2\) (as in the definitions in equation (2.68)); the \(\lambda_{ix,k*}\) are related to the maximum eigenvalues of \(J_f\) and \(J_h\) at the indicated grid points: \(\lambda_r = |u_r| + c\) and \(\lambda_z = |w| + c\). The scaling parameters \(\kappa^{2,4}\) have the nominal values \(1/4\) and \(1/256\), as suggested in Ref. 42, but will be further scaled to improve simulation results.

The most influential term in equation (2.68) is \(\nu_{i,k}\):

\[
\nu_{i,k} = \frac{|p_{i+1,k} - 2p_{i,k} + p_{i-1,k}|}{|p_{i+1,k} + 2p_{i,k} + p_{i-1,k}|}
\]

(2.69)

which is related to the pressure gradient in the flow. Note that for finite differences along the streamwise direction, \(\nu_{i,k}\) is evaluated by taking differences with respect to \(k\) rather than \(i\). For large \(\nu\), such as in the presence of shock waves, fourth-order dissipation “turns off” and second-order dissipation dominates. In contrast, the second-order term can not suppress (non-physical) oscillations in regions of smooth flow, necessitating the higher-order dissipation term.

From equation (2.67), it can be seen that each highly-nonlinear damping term is multiplied by a finite-difference of \(u_{i,k}\), suggesting that damping matrices can be constructed for each term. To maintain notational consistency with \(\epsilon^{2,4}\) and \(\kappa^{2,4}\) in
equation (2.68), these matrices are defined as $P^2(x)$ and $P^4(x)$, from which a modified state matrix can be constructed as

$$\dot{x} = A(x)x + B(x)u + P^2(x)x - P^4(x)x$$

$$= (A(x) + P^2(x) - P^4(x))x + B(x)u$$

(2.70)

$$= \dot{A}(x)x + B(x)u$$

**Constructing the Second-Order Damping Matrix.** The scaling factor $\epsilon^2$ is calculated from a four-point stencil on the isolator grid, requiring a boundary condition treatment that considers grid points at the edges of the grid ($i = 1$, $Nr$ and $k = 1$, $Nz$) and those points just inside the edges ($i = 2$, $Nr - 1$ and $k = 2$, $Nz - 1$). Before addressing the boundary conditions, the general structure of the block row of $P^2$ is found by expanding the second-order damping term in equation (2.67):

$$(D^2_r + D^2_z)u_{i,k} = \{\nabla_r [\lambda_{i+0.5,k}\epsilon^2_{i+0.5,k}\Delta_r] + \nabla_z [\lambda_{i,k+0.5}\epsilon^2_{i,k+0.5}\Delta_z]\} u_{i,k}$$

$$= \lambda_{i+0.5,k}\epsilon^2_{i+0.5,k}u_{i+1,k} + \lambda_{i,k+0.5}\epsilon^2_{i,k+0.5}u_{i,k+1} + \lambda_{i-0.5,k}\epsilon^2_{i-0.5,k}u_{i-1,k} + \lambda_{i,k-0.5}\epsilon^2_{i,k-0.5}u_{i,k-1}$$

$$- [\lambda_{i+0.5,k}\epsilon^2_{i+0.5,k} + \lambda_{i,k+0.5}\epsilon^2_{i,k+0.5} + \lambda_{i-0.5,k}\epsilon^2_{i-0.5,k} + \lambda_{i,k-0.5}\epsilon^2_{i,k-0.5}]u_{i,k}$$

$$= \alpha_{i+1,k}u_{i+1,k} + \alpha_{i,k+1}u_{i,k+1} + \alpha_{i-1,k}u_{i-1,k} + \alpha_{i,k-1}u_{i,k-1} - \alpha^i_ku_{i,k}$$

(2.71)

where $\alpha_{i*,k*} = \lambda_{i*,k*}\epsilon^2_{i*,k*}$, with $i^*$ and $k^*$ being a backwards or forward index shift, and $\alpha^i_k$ is the sum of these coefficients.

**Boundary Condition Treatment at Grid Edge.** Boundary condition treatments for implementing damping in CFD codes are presented by Pulliam in Ref. 41 and Swanson and Turkel in Ref. 40, but are limited in application with this model because it is not a CFD solver. The boundary condition treatment developed here is adapted from these suggestions and designed to reflect that implemented in constructing $A(x)$, since boundary conditions replace the governing equations at the grid edges.
In the radial direction, the governing equations at \( i = 1 \) and \( i = N_r \) are discretized using a single finite-difference. In a similar manner, calculation of the damping coefficients \( \alpha_{i\pm1,k} \) are modified to remove dependence on points outside of the grid:

\[
\alpha_{i+1,k}^* = \begin{cases} 
\lambda_{i+0.5,k} \kappa^2 \nu_{i+1,k} & \text{for } i = 1 \\
0 & \text{for } i = N_r
\end{cases} \\
\alpha_{i-1,k}^* = \begin{cases} 
0 & \text{for } i = 1 \\
\lambda_{i-0.5,k} \kappa^2 \nu_{i-1,k} & \text{for } i = N_r
\end{cases}
\] (2.72)

When the finite-difference stencil extends beyond the isolator grid, \( \alpha_{i\pm1,k}^* = 0 \). Notice that when \( \alpha_{i\pm1,l}^* = 0 \), calculation of \( \epsilon_{i\pm1,k}^2 \) does not require evaluation of a maximum because the three-point stencil required to calculate \( \nu_{i,k} \) involves a point outside of the grid. Equation (2.72) preserves the definition of \( \alpha_{k}^i \), which multiplies \( u_{i,k} \) in equation (2.71).

A similar approach is taken in the streamwise direction. At the isolator entrance plane, the flow is supersonic and boundary conditions are without dependence on interior points of the grid. At \( k = 1 \), \( \alpha_{i,1} \) and \( \alpha_{i,2} \) vanish and there is no damping. For simplicity, this same scheme was implemented at \( k = N_z \) (\( \alpha_{i,N_z} = \alpha_{i,N_z-1} = 0 \)), but a velocity-dependent scheme that would be more reflective of the treatment implemented in constructing \( A(x) \) may be implemented if necessary. This would increase the complexity of the model while reducing local errors introduced by the simpler boundary condition treatment.

**Boundary Condition Treatment at First Interior Point.** A similar boundary condition treatment can be implemented for grid points just inside the boundary \((i = 2, N_r - 1 \text{ and } k = 2, N_z - 1)\). For each grid point \((i, k)\) in these regions, calculation
of $\alpha_{i\pm1,k}$ or $\alpha_{i,k\pm1}$ depends on the location of the point in the grid:

$$\alpha^{*}_{i+1,k} = \begin{cases} 
\alpha_{i+1,k} & \text{for } i = 2 \\
\lambda_{i+0.5,k}^{2}\nu_{i,k} & \text{for } i = N_{r} - 1
\end{cases}$$

$$\alpha^{*}_{i-1,k} = \begin{cases} 
\lambda_{i-0.5,k}^{2}\nu_{i,k} & \text{for } i = 2 \\
\alpha_{i-1,k} & \text{for } i = N_{r} - 1
\end{cases}$$

$$\alpha^{*}_{i,k+1} = \begin{cases} 
\alpha_{i,k+1} & \text{for } k = 2 \\
\lambda_{i,k+0.5,k}^{2}\nu_{i,k} & \text{for } k = N_{z} - 1
\end{cases}$$

$$\alpha^{*}_{i,k-1} = \begin{cases} 
\lambda_{i,k-0.5,k}^{2}\nu_{i,k} & \text{for } k = 2 \\
\alpha_{i,k-1} & \text{for } k = N_{z} - 1
\end{cases}$$

Like equation (2.72), a maximum is not evaluated when the calculation of $\epsilon_{i\pm1,k}^{2}$ or $\epsilon_{i,k\pm1}^{2}$ depends on a point outside of the grid. For example, when $i = 2$, $\nu_{i\pm1,k}$ would not be calculated and the maximum function would be replaced with $\nu_{i,k}$. Again, equation (2.73) does not affect the definition of $\alpha^{i}_{k}$.

Like $A(x)$, $P^{2}(x)$ has a block tridiagonal structure with off-diagonal blocks that are diagonal matrices (equations (2.74) and (2.75)) and diagonal blocks that have tridiagonal structure (equation (2.76)):

$$P^{2}_{sub,i}(x) = \begin{bmatrix} 
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \alpha_{i-1,2} & 0 & \ldots & 0 & 0 \\
0 & 0 & \alpha_{i-1,3} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \alpha_{i-1,N_{z}-1} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}$$

(2.74)
Note the modifications to $P_{\text{sub},2}^2(x)$, $P_{\text{sub},N_r}^2(x)$, $P_{\text{sup},1}^2(x)$ and $P_{\text{sup},N_r-1}^2(x)$ made necessary by the boundary condition treatments in equations (2.72) and (2.73). Also notice that the diagonal terms of $P_{d,i}^2(x)$ are equal to the sum of all of the damping coefficients in the $i^{th}$ block row; in most cases $\alpha_{i,k}^i = \alpha_{i+1,k} + \alpha_{i-1,k} + \alpha_{i,k+1} + \alpha_{i,k-1}$, with boundary modifications as previously described.

The block structure of $P^2(x)$ can be seen in equation (2.77), where submatrices with an "*" have been modified following the boundary condition treatment. Note that each submatrix contains blocks that are 4-by-4 identity matrices scaled by $\alpha_{i,k}$ (or $\alpha_{k}^i$), so that $P^2(x)$ is a banded matrix, with five bands symmetric about the main
diagonal

\[ P^2(x) = \begin{bmatrix}
P^2_{d,1} & P^2_{\text{sup},1} & 0 & \ldots & 0 & 0 \\
P^2_{\text{sub},2} & P^2_{d,2} & P^2_{\text{sup},2} & \ldots & 0 & 0 \\
0 & P^2_{\text{sub},3} & P^2_{d,3} & P^2_{\text{sup},3} & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \ldots & P^2_{\text{sub},N_r} & P^2_{d,N_r-1} & P^2_{\text{sup},N_r-1} \\
0 & 0 & \ldots & \ldots & P^2_{\text{sub},N} & P^2_{d,N} \\
\end{bmatrix} \quad (2.77) \]

Building the Fourth-Order Damping Matrix \( P^4 \). The additional differencing operations in the fourth-order damping term produce a more complex expression than equation (2.71):

\[
(D_r^4 + D_z^4)u_{i,k} = \lambda_i + 0.5, k \epsilon_{i+0.5,k}^4 \epsilon_{i+0.5,k} \left( u_{i+2,k} - 3u_{i+1,k} + 3u_{i,k} - u_{i-1,k} \right) \\
- \lambda_{i-0.5, k} \epsilon_{i-0.5,k}^4 \left( u_{i+1,k} - 3u_{i,k} + 3u_{i-1,k} - u_{i-2,k} \right) \\
+ \lambda_{i,k+0.5} \epsilon_{i+0.5,k+0.5} \left( u_{i,k+2} - 3u_{i,k+1} + 3u_{i,k} - u_{i,k-1} \right) \\
- \lambda_{i,k-0.5} \epsilon_{i-0.5,k}^4 \left( u_{i,k+1} - 3u_{i,k} + 3u_{i,k-1} - u_{i,k-2} \right) \\
= \lambda_i + 0.5, k \epsilon_{i+0.5,k}^4 \epsilon_{i+0.5,k} \left( u_{i+2,k} - \left( \lambda_{i-0.5, k} \epsilon_{i-0.5,k}^4 + 3\lambda_{i+0.5, k} \epsilon_{i+0.5,k}^4 \right) u_{i+1,k} \\
- (3\lambda_{i-0.5, k} \epsilon_{i-0.5,k}^4 + \lambda_{i+0.5, k} \epsilon_{i+0.5,k}^4) u_{i-1,k} + \lambda_{i-0.5, k} \epsilon_{i-0.5,k}^4 u_{i-2,k} \\
+ \lambda_{i,k+0.5} \epsilon_{i+0.5,k+0.5} u_{i,k+2} - (\lambda_{i,k-0.5} \epsilon_{i-0.5,k}^4 + 3\lambda_{i,k+0.5} \epsilon_{i+0.5,k+0.5}) u_{i,k+1} \\
- (3\lambda_{i,k-0.5} \epsilon_{i-0.5,k}^4 + \lambda_{i,k+0.5} \epsilon_{i+0.5,k+0.5}) u_{i,k-1} + \lambda_{i,k-0.5} \epsilon_{i-0.5,k}^4 u_{i,k-2} \\
+ 3(\lambda_{i+0.5, k} \epsilon_{i+0.5,k}^4 \epsilon_{i+0.5,k} + \lambda_{i-0.5, k} \epsilon_{i-0.5,k}^4 \epsilon_{i-0.5,k}) \\
+ \lambda_{i,k-0.5} \epsilon_{i-0.5,k}^4 \epsilon_{i-0.5,k} \right) u_{i,k} \\
= \beta_{i+1,k} u_{i+2,k} - (3\beta_{i+1,k} + \beta_{i-1,k}) u_{i+1,k} - (\beta_{i+1,k} + 3\beta_{i-1,k}) u_{i-1,k} \\
+ \beta_{i-1,k} u_{i-2,k} + \beta_{i,k+1} u_{i,k+2} - (3\beta_{i,k+1} + \beta_{i,k-1}) u_{i,k+1} \\
- (\beta_{i,k+1} + 3\beta_{i,k-1}) u_{i,k-1} + \beta_{i,k-2} u_{i,k-2} + \beta_{i,k} u_{i,k} \]

Here, \( \beta_{i+k} = \lambda_{i+k} \epsilon_{i+k}^4 \epsilon_{i+k} \) and \( \beta_k \) is the sum of all coefficients on \( u_{i+2,k} \), \( u_{i+1,k} \), \( u_{i,k+2} \), and \( u_{i,k+1} \), similar to the definitions of \( \alpha_{i+k} \) and \( \alpha_k \) in the second-order damping
term. For a general point \((i,k)\), equation (2.78) holds, but boundary conditions at 
\(i = 1, 2, N_r - 1, N_r\) and \(k = 1, 2, N_z - 1, N_z\) simplify the damping terms at these points.

**Boundary Condition Treatment at Grid Edge.** Pulliam suggests two ways to modify the model at the boundaries [41]: replace the five-point central difference with either a lower-order approximation (three-point central difference), introducing local error, or a non-centered 5-point scheme. For simplicity, the former modification is introduced, producing a second-order damping term for points at the edges of the grid:

\[
-(D_r^4 + D_z^4) u_{i,k} = - \{ \nabla_r \left[ \lambda_{i+0.5,k} \epsilon_{i+0.5,k} \Delta_r \right] + \nabla_z \left[ \lambda_{i,k+0.5} \epsilon_{i,k+0.5} \Delta_z \right] \} u_{i,k}
\]

\[
(D_r^4 + D_z^4) u_{i,k} = \lambda_{i+0.5,k} \epsilon_{i+0.5,k} (u_{i+1,k} - u_{i,k}) - \lambda_{i-0.5,k} \epsilon_{i-0.5,k} (u_{i,k} - u_{i-1,k})
\]

\[
+ \lambda_{i,k+0.5} \epsilon_{i,k+0.5} (u_{i,k+1} - u_{i,k}) - \lambda_{i,k-0.5} \epsilon_{i,k-0.5} (u_{i,k} - u_{i,k-1})
\]

\[
= \lambda_{i+0.5,k} \epsilon_{i+0.5,k} u_{i+1,k} + \lambda_{i-0.5,k} \epsilon_{i-0.5,k} u_{i-1,k}
\]

\[
+ \lambda_{i,k+0.5} \epsilon_{i,k+0.5} u_{i,k+1} + \lambda_{i,k-0.5} \epsilon_{i,k-0.5} u_{i,k-1}
\]

\[
- (\lambda_{i+0.5,k} \epsilon_{i+0.5,k} + \lambda_{i-0.5,k} \epsilon_{i-0.5,k} + \lambda_{i,k+0.5} \epsilon_{i,k+0.5} + \lambda_{i,k-0.5} \epsilon_{i,k-0.5}) u_{i,k}
\]

\[
= \beta_{i+1,k}^* u_{i+1,k} + \beta_{i-1,k}^* u_{i-1,k} + \beta_{i,k+1}^* u_{i,k+1}
\]

\[
- \beta_{i,k-1}^* u_{i,k-1} + \beta_i^* u_{i,k}
\]

Note that here \(\beta_i^* = \beta_{i+1,k}^* + \beta_{i-1,k}^* + \beta_{i,k+1}^* + \beta_{i,k-1}^*\), preserving the definition of \(\beta_k^*\) at the boundary points. Because this expression is similar to the second-order dissipation term, the boundary condition treatment implemented for that term is extended here.
At the grid boundary, in the radial direction, equation (2.79) can be simplified because only one finite-difference can be calculated when \( i = 1 \) or \( i = N_r \):

\[
\beta^*_{i+1,k} = \begin{cases} 
\lambda_{i+0.5,k} \max\{0, \kappa^4 - \kappa^2 \nu_{i+1,k}\} & \text{for } i = 1 \\
0 & \text{for } i = N_r 
\end{cases}
\]

\[
\beta^*_{i-1,k} = \begin{cases} 
\lambda_{i-0.5,k} \max\{0, \kappa^4 - \kappa^2 \nu_{i-1,k}\} & \text{for } i = N_r \\
0 & \text{for } i = 1 
\end{cases}
\]

\[ (2.80) \]

Calculation of \( \epsilon^2_{i\pm1,k} \) only involves calculation of \( \nu_{i\pm1,k} \), since \( \nu_{i,k} \) depends on points outside the grid. And the definitions in equation (2.80) are made to preserve the definition of \( \beta^*_k \).

In the streamwise direction, \( \beta_{i,1} = \beta_{i,2} = \beta_{i,N_z-1} = \beta_{i,N_z} = 0 \) to reflect the boundary condition treatment implemented for \( P^2(x) \), where there is no damping at the isolator entrance and exit planes.

**Boundary Condition Treatment at First Interior Point.** For grid points with indices \( i = 2, \ N_r - 1 \) and \( k = 2, \ N_z - 1 \), equation (2.79) holds with the following definitions:

\[
\beta^*_{i+1,k} = \begin{cases} 
\beta_{i+1,k} & \text{for } i = 2 \\
\lambda_{i+0.5,k} \max\{0, \kappa^4 - \kappa^2 \nu_{i,k}\} & \text{for } i = N_r - 1 
\end{cases}
\]

\[
\beta^*_{i-1,k} = \begin{cases} 
\lambda_{i-0.5,k} \max\{0, \kappa^4 - \kappa^2 \nu_{i,k}\} & \text{for } i = 2 \\
\beta_{i-1,k} & \text{for } i = N_r - 1 
\end{cases}
\]

\[
\beta^*_{i,k+1} = \begin{cases} 
\beta_{i,k+1} & \text{for } k = 2 \\
\lambda_{i,k+0.5} \max\{0, \kappa^4 - \kappa^2 \nu_{i,k}\} & \text{for } k = N_z - 1 
\end{cases}
\]

\[
\beta^*_{i,k-1} = \begin{cases} 
\lambda_{i,k-0.5} \max\{0, \kappa^4 - \kappa^2 \nu_{i,k}\} & \text{for } k = 2 \\
\beta_{i,k-1} & \text{for } k = N_z - 1 
\end{cases}
\]

\[ (2.81) \]
where modifications in the second-order damping terms are present in the arguments of the maximum functions. Again, the definition of $\beta_k^i$ does not change for these grid points.

Due to the extended stencil for approximating the derivative in the fourth-order damping term, $P^4(x)$ has a block-pentadiagonal structure, with submatrices on the sub-subdiagonals and super-superdiagonals (equations (2.82) and (2.83)). Because of the boundary condition treatment, these matrices are only constructed for $i \in [3 N_r - 2]$ and $k \in [3 N_z - 2]$. On this same index range, the sub-diagonal and super-diagonal matrices, equations (2.84) and (2.85), are diagonal matrices with linear combinations of $\beta_{i\pm1,k}$ along the diagonal. The diagonal blocks of $P^4(x)$ have a pentadiagonal structure, as can be seen from equation (2.86):

\[
P_{\text{sub}2,i}^4(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \beta_{i-1,2} & 0 & \ldots & 0 & 0 \\
0 & 0 & \beta_{i-1,3} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \beta_{i-1,N_z-1} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}
\quad \text{for } i < N_r - 1 \quad (2.82)
\]

\[
P_{\text{sup}2,i}^4(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \beta_{i+1,2} & 0 & \ldots & 0 & 0 \\
0 & 0 & \beta_{i+1,3} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \beta_{i+1,N_z-1} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}
\quad \text{for } i > 2 \quad (2.83)
\]
\[ P^{l}_{sub,i}(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \beta_{i-1,2} & 0 & \ldots & 0 & 0 \\
0 & 0 & \beta_{i-1,3} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \beta_{i-1,N_z-1} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 \\
\end{bmatrix} \quad \text{for } i = 2, N_r - 1, N_r \]

\[ P^{l}_{sub,i}(x) = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & -(\beta_{i+1,2} + 3\beta_{i-1,2}) & 0 & \ldots & 0 \\
0 & 0 & -(\beta_{i+1,3} + 3\beta_{i-1,3}) & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 \\
\end{bmatrix} \quad \text{otherwise} \]

(2.84)
\begin{align*}
P^4_{\text{sup},i}(x) &= \begin{bmatrix} 
0 & 0 & 0 & \ldots & 0 & 0 \\
0 & \beta_{i+1,2} & 0 & \ldots & 0 & 0 \\
0 & 0 & \beta_{i+1,3} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \beta_{i+1,N_z-1} & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 
\end{bmatrix} \\
&= \begin{bmatrix} 
0 & 0 & 0 & \ldots \\
0 & -(3\beta_{i+1,2} + \beta_{i-1,2}) & 0 & \ldots \\
0 & 0 & -(3\beta_{i+1,3} + \beta_{i-1,3}) & \ldots \\
\vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & \ldots \\
0 & 0 & 0 & \ldots 
\end{bmatrix} \\
\text{for } i = 1, 2, N_r - 1
\end{align*}

\text{(2.85)}

\text{otherwise}
\[
P^i_{d,i}(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & \ldots \\
\beta_{i,1}^* & -\beta_2^i & \beta_{i,3} & 0 & 0 & \ldots \\
\beta_{i,2}^* & -(\beta_{i,4} + 3\beta_{i,2}) & -\beta_3^i & -(3\beta_{i,4} + \beta_{i,2}) & \beta_{i,4} & \ldots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & \beta_{i,N_z-3}^* & -(\beta_{i,N_z-1} + 3\beta_{i,N_z-3}) & -\beta_{N_z-2}^i & \beta_{i,N_z-2} & \ldots \\
0 & \ldots & 0 & 0 & \beta_{i,N_z-2} & \ldots & \ddots \\
0 & \ldots & 0 & 0 & 0 & \ldots & \ldots \\
\ldots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\ldots & -(3\beta_{i,N_z-1} + \beta_{i,N_z-3}) & \beta_{i,N_z-1} & \beta_{i,N_z}^* & \beta_{i,N_z} & \ddots \\
\ldots & -\beta_{N_z-1}^i & \beta_{i,N_z} & \ddots & \ddots & \ddots & \ddots \\
\ldots & 0 & 0 & \ddots & \ddots & \ddots & \ddots \\
\end{bmatrix}
\]

Note that the diagonal elements in equations (2.84) and (2.85) are modified at the boundary points and the sub-subdiagonal and super-superdiagonal submatrices are zero outside the index range indicated in equations (2.82) and (2.83).

Unlike \(\alpha_k^i\), \(\beta_k^i\) is not always the sum of damping coefficients \(\beta_{i\pm1,k}\) and \(\beta_{i,k\pm1}\) because of the fourth-derivative terms. A summary of the calculation of the diagonal term is as follows:

- \(i = 1, N_r\) and \(k \in (2N_z - 1)\): \(\beta_k^i = \beta_{i\pm1,k} - 3(\beta_{i,k-1} + \beta_{i,k+1})\)
- \(i = 2, N_r - 1\) and \(k \in (2N_z - 1)\): \(\beta_k^i = \beta_{i+1,k} + \beta_{i-1,k} - 3(\beta_{i,k-1} + \beta_{i,k+1})\)
- \(i \in (2N_r - 1)\) and \(k = 2, N_z - 1\): \(\beta_k^i = -3(\beta_{i+1,k} + \beta_{i-1,k}) + \beta_{i,k+1} + \beta_{i,k-1}\)
- \(i \in (2N_r - 1)\) and \(k \in (2N_z - 1)\): \(\beta_k^i = -3(\beta_{i+1,k} + \beta_{i-1,k} + \beta_{i,k+1} + \beta_{i,k-1})\)
This can be summarized as “$\beta_{i,k} = \text{the sum of the coefficients of the ‘shifted’ state variables } u_{i,k}$,” as emphasized throughout the discussion.

The damping matrix $P^4(x)$ has the block pentadiagonal structure shown in equation (2.87), where boundary modifications are indicated by an asterisk next to the submatrix. Note that the elements of each submatrix are 4-by-4 identity matrices scaled by $\beta_{i,k}$ (or $\beta_{i,k}^2$), so that $P^4(x)$ is a banded matrix with nine bands located symmetrically about the main diagonal.

$$P^4(x) = \begin{bmatrix}
p^4_{d,1} & p^4_{sup,1} & 0 & 0 & 0 & \ldots & 0 
p^4_{sub,2} & p^4_{d,2} & p^4_{sup,2} & 0 & 0 & \ldots & 0 
p^4_{sub,3} & p^4_{d,3} & p^4_{sup,3} & p^4_{sup,2,3} & \ldots & 0 
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 
0 & \ldots & 0 & p^4_{sup,2,N_r-2} & p^4_{sup,N_r-2} & p^4_{d,N_r-2} & p^4_{sup,N_r-2} & \ldots & 0 
0 & \ldots & 0 & 0 & p^4_{sup,N_r-1} & p^4_{d,N_r-1} & p^4_{sup2,N_r-2} & p^4_{sup,N_r-2} & \ldots & 0 
0 & \ldots & 0 & 0 & 0 & \ldots & 0 & p^4_{sup,N_r} & p^4_{d,N_r} & \ldots & 0
\end{bmatrix}$$

(2.87)

It was found during validation that the nominal values of $\kappa^{2,4}$ (that is, $\kappa^2 = 1/4$ and $\kappa^4 = 1/256$) had little effect on the behavior of the system, a conclusion supported by the observation that $A(\bar{x})$ and $\hat{A}(\bar{x})$ has similar eigenvalues, suggesting the linearized models would have similar dynamics. A closer look at the elements of $A(x)$, $P^2(x)$ and $P^4(x)$ revealed that the order of magnitude of elements in the damping matrices was much smaller than those in $A(x)$, suggesting the effects of damping could be increased by tuning the damping coefficients that make up $P^2(x)$ and $P^4(x)$. A scaling factor for each coefficient $\alpha_{i,k}$ and $\beta_{i,k}$ was introduced and tuned so that the order of magnitude of $P^2(x)$ and $P^4(x)$ more closely matched that of $A(x)$. This will be further addressed in the discussion of simulation results in the next section.

**Model Validation**

The isolator model was implemented as an s-function in SIMULINK® in order to be able to monitor the time-progress of the simulation and ensure data would not be
lost in case of termination. This was an important feature because simulation of the nonlinear models took much longer to run than simulation of the linearized model and were stopped early to evaluate results.

After simulating the undamped model (equation (2.66)), several simulations of the model with damping were run to tune the scaling factors and evaluate the effects of including damping. Coefficients on $\alpha_{i\pm1,k}$ and $\alpha_{i,k\pm1}$ were tuned through simulation of the model with only second-order damping implemented and an attempt to tune coefficients on $\beta_{i\pm1,k}$ and $\beta_{i,k\pm1}$ through simulation of the model with both second- and fourth-order damping. It will be shown that adequate scaling was not achieved because simulations could not be efficiently run despite optimizing the code for use with the Parallel Computing Toolbox (PCT) in MATLAB®.

**Code optimization.** The code for simulation of the nonlinear isolator model was related to the code used to construct the SPM, which employed nested for-loops to calculate the system matrices. For the SPM, the run-time of this code did not present an obstacle because it was run only once in constructing the model; for simulation of the nonlinear model, however, the matrices are evaluated at each time-step, creating a bottleneck in simulation. MATLAB® provides a ‘code profiler’ that analyzes code and identifies which lines require the highest computational cost. It was found that in most cases, these lines were in the function written to construct $A(x)$, so code optimization focused on this function.

Hardware limitations prohibited the use of the full matrix data type in constructing $A(x)$ and the portions of code related to construction and manipulation of the sparse matrix were found to be most responsible for the long run-time of the function. Additional improvements were achieved by reducing the number of loops and replacing built-in MATLAB® functions, such as the dot product, with one or two lines of code accomplishing the same task. The code was modified in five steps, through
replacement of direct manipulation of the sparse matrix $A(x)$ with calls to `spalloc`, `spdiags`, constructing the block-diagonal portions of $A(x)$ and $C(x)$, and replacing the use of `dot` with a direct calculation. After these modifications were made, nested for-loops were replaced with a single parfor-loop, so that the PCT can be used, when available.

The nonlinear isolator model was simulated on a Dell Precision T1500 workstation running 64-bit Windows 7 Enterprise with 16 GB of RAM and a 2.80 GHz quad-core Intel Core i5 processor. Simulations were run in 64-bit MATLAB®, with the PCT available. Opening a pool of four MATLAB® ‘workers’ (one ‘worker’ per processor core) allows iterations of the parfor-loops to be distributed to the workers, speeding up the code. Utilizing the PCT, the time to construct the SPM was improved by a factor of 122.093, reducing the run-time per time-step from 346.133 sec to 2.835 sec. Unfortunately, further optimization is limited by indexing rules in the parfor-loop structure.

**Simulation Results and Analysis.** One of the biggest drawbacks of this model is that $A(x)$, $B(x)$, $P^2(x)$ and $P^4(x)$ must be calculated at each time step. Even with code optimization, it was not possible to obtain sufficient simulation data for adequate analysis of the model with damping implemented. The results presented here are from simulations that have been terminated after running for two weeks. (The simulations could be allowed to run for a longer time if more data were desired.)

Results from each simulation are presented in three figures. Pressure profiles along the isolator wall and center-line are presented in the first figure, where it is assumed that pressure at each coordinate is of interest. These plots allow for observation of shock propagation due an the increase of backpressure from $p_{bA} = 218.577$ kPa to $p_{bC} = 226.313$ kPa (an increase of 3.54%). The shock wave is expected to move upstream from 0.86 m to 0.66 m, although simulations were not run for enough time.
to verify that the model captures this propagation. The other two figures show profiles of $\rho$, $u_r$, $w$, and $E$ along the center-line and wall, respectively. These plots provide insight into the behavior of the pressure profiles, where it is possible to identify which flow variables may contribute to inaccuracies in the simulation.

**Simulation of the Undamped Model.** The first simulation was run for the undamped nonlinear model in equation (2.66). The simulation was terminated after running for two weeks, during which 1.32 ms of data was collected. At the end of this simulation, the shock wave appears to be propagating upstream as the pressure that had built up downstream of the shock begins to dissipate. Plots of the flow variables, however, suggest that the flow dynamics would not be accurately modeled if the simulation were run for a longer time to verify the shock motion.

The build up of pressure downstream of the shock can clearly be seen from the pressure profiles in Figure 2.16. After 1.25 ms, the shock wave propagates upstream as the downstream pressure dissipates, suggesting that damping may not be necessary if the initial millisecond of data is transient behavior attributed to numerical errors the model. However, a closer look at the flow variables provides support for implementing damping.

The dynamics of the flow variables can be seen from the flow variable profiles at the center-line and wall of the isolator (Figures 2.17 and 2.18). In each plot, the shock location, and apparent shock motion at the end of the simulation, can be seen.

Considering first the flow dynamics along the center-line in Figure 2.17, it can be seen that the order of magnitude of each flow variable increases significantly. The radial velocity, which is relatively small at the start of the simulation, increases by nearly two orders of magnitude downstream of the shock. In this same region, streamwise velocity exhibits a similar increase and just downstream of the entrance plane the order of magnitude of $w$ changes sharply, decreasing toward zero before
Figure 2.16: Pressure profiles along the center-line (top) and wall (bottom) of the isolator, found from simulation of the undamped model.

changing direction soon after the simulation starts. This is an undesirable and unexpected occurrence requiring resolution through further examination of the model. This undesirable behavior manifests itself in the density profiles through a sharp decrease toward 0 near the entrance of the isolator. Like radial velocity, internal energy sharply increases (nearly three orders of magnitude) downstream of the shock, which is clearly visible in the profiles.

While the general shape of the profiles at the wall, in Figure 2.18, are similar to those at the center-line, two things are interesting to note. The first is that the radial velocity increases by three orders of magnitude downstream of the shock, as compared to two at the center-line. This may be related to the second observation: the initial conditions of each variable at the wall are slightly different than those at the center-line. The CFD data used to construct the initial conditions exhibit some anomalies in the center-line data, such as spikes in the velocity profiles at discrete
points along the center-line, which can be ‘smoothed out’ by tuning the second-order damping parameters. Tuning these parameters produces flow-fields that are radially smooth, as observed in the CFD data; if experimental data were available, the model could instead be tuned to produce results reflective of this alternative set of data (which is not guaranteed to be radially smooth).

**Simulation of the Model with Second-Order Damping.** The model with the second-order damping scheme (equation (2.70) with $\tilde{A}(x) = A(x) + P^2(x)$) was simulated next. The inclusion of $P^2(x)$ had a noticeable effect on the dynamics of both the output (pressure) and the flow variables (density, velocity and internal energy), but not on the shock-capturing capability of the model. Another drawback to this model was that computation time was increased from the presence of $P^2(x)$ and only 0.31 ms of data could be collected in two weeks, which is not a sufficient amount to adequately evaluate the model.
Figure 2.18: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the wall of the isolator, from simulation of the undamped model.

It should be noted that the damping scheme implemented in this simulation contains a component-wise check of the velocity, so that damping is implemented in the radial (streamwise) direction only when $u_r$ ($w$) is subsonic. This limits the adverse effects of the damping in regions of supersonic flow (upstream of the shock) [41].

The limited shock capturing capabilities of this model are highlighted in Figure 2.19, pressure profiles at the center-line and wall of the isolator (the top and bottom subplots, respectively). The most notable feature of these profiles is that the shock appears to gradually propagate downstream, the reverse of what is expected. Despite this, the pressure profiles show two important improvements over those in Figure 2.16: the pressure build-up downstream of the shock was mitigated by the presence of damping and the shape of the profiles was improved. Downstream of the shock, pressure profiles do not degrade as they do in Figure 2.16, and the radial anomalies appear to be smoothed out by the presence of damping. And although
Figure 2.19: Pressure profiles along the center-line (top) and wall (bottom) of the isolator from simulation of the model with second-order damping ($\hat{A}(x) = A(x) + P^2(x)$).

Pressure does not tend to zero upstream of the shock, as in the undamped model, this degradation can be expected if the simulation were run for a longer time.

The effect of second-order damping can be further observed through comparison of the flow variable profiles from the undamped and damped models. Figures 2.20 and 2.21 show these profiles for simulation of the model with second-order damping, from which the effects of tuning the damping parameters can be observed by specifically considering a single flow variable, such as $w$. The anomalies in the initial conditions can be seen by comparing the dark red lines in the center-line and wall profiles. Since the flow is expected to be radially-smooth, the damping parameters related to discretization in the radial direction were tuned achieve the smooth flow profiles. It should be noted that, along the center-line each flow variable exhibited a rapid increase in magnitude at the exit plane of the isolator that was not mitigated...
Simulation Results: Flow Variables at Centerline

Figure 2.20: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the center-line of the isolator, from simulation of the model with second-order damping.

by the damping because of the boundary condition treatment. (This is seen most clearly in the plots of radial velocity and internal energy in Figure 2.20.)

Since the flow variables in each location behave similarly due to the damping, further observation can be made by considering the profiles at the wall (Figure 2.21) since these more closely match the rest of the flow. The observations made of the pressure profiles in Figure 2.19 can be transferred to the profiles of the flow variables at the wall, which maintain the expected profile shape better than in simulation of the undamped system (Figure 2.16), despite the slight degradation observed in the internal energy profile. It should also be noted that, although $w$ remains positive and $\rho$ does not decrease as much for this simulation as observed for simulation of the undamped model, this is only due to the fact that only one-quarter of the amount of data was collected. If simulation of the model with second-order damping was allowed to run for a longer time, these undesirable occurrences are expected to be observed.
Simulation Results: Flow Variables at Wall

Figure 2.21: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the wall of the isolator, from simulation of the model with second-order damping.

Simulation of the Model with Second- and Fourth-Order Damping. The final simulations were run for the model with second- and fourth-order damping implemented. The model was first simulated with all damping implemented only in regions of subsonic flow (component-wise). In the second simulation, damping was implemented more closely to a scheme discussed by Pulliam [41], where second-order damping is implemented everywhere and fourth-order damping only in the subsonic regions. Differences between these implementations can not be seen, however, due to the small amount of data collected.

Results from simulation of the system with second- and fourth-order damping in subsonic regions only are shown in Figures 2.22 to 2.24. MATLAB® encountered an error that terminated this simulation after one week, so that only 0.19 ms of data was collected (this error was reproduceable, but it was not possible to determine the
Figure 2.22: Pressure profiles along the center-line (top) and wall (bottom) of the isolator from simulation of the model with second- and fourth-order damping ($\hat{A}(x) = A(x) + P^2(x) - P^4(x)$).

The final set of results, in Figures 2.25 to 2.27, are from simulation of the model with second-order damping implemented in the entire flow domain and fourth-order damping implemented only in subsonic regions. Again, this simulation was run for just one week due to errors encountered in MATLAB®, in which time 0.2 ms of data was gathered. This is not enough data from which to draw conclusions about the accuracy of the model or the implemented damping scheme, but the results are included here for completeness.

Conclusions

The large-scale nonlinear model of an axisymmetric scramjet isolator constructed following the methodology presented by Chicatelli and Hartley [28,29] was found to be
Figure 2.23: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the center-line of the isolator, from simulation of the model with second- and fourth-order damping.

Figure 2.24: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the wall of the isolator, from simulation of the model with second- and fourth-order damping.
Figure 2.25: Pressure profiles along the center-line (top) and wall (bottom) of the isolator from simulation of the model with second-order damping everywhere, and fourth-order damping in regions of subsonic flow.

Figure 2.26: Flow variable profiles (velocity components (top), density and internal energy (bottom)) at the isolator center-line, from simulation of the model with second-order damping everywhere and fourth-order damping in subsonic flow regions.
Simulation Results: Flow Variables at Wall

Figure 2.27: Flow variable profiles (velocity components (top), density and internal energy (bottom)) along the wall of the isolator, from simulation of the model with second-order damping everywhere, and fourth-order damping in subsonic flow regions.

an inefficient model for use in a model-based control design methodology. The state vector of the nonlinear model had \( n = 29204 \), so construction of the many \( n \)-by-\( n \) state-dependent matrices at each time step slowed simulations significantly in relation to the SPM. Despite implementation on a computer with a multi-core processor, to take advantage of the PCT, only 1.25 ms of data could be collected in two weeks of running the simulation of the undamped model. The increase in computational cost associated with the need to evaluate three \( n \)-by-\( n \) matrices at each time step, accompanied by software problems, further limited the amount of data collected for simulations of the model with damping. The small amount of data was not enough to draw useful conclusions about the model or the effects of damping, and it was nearly impossible to tune the model due to the slow turn-around time.

Since the amount of data collected should be balanced with the amount of time it takes to collect the data, a smaller-order nonlinear model should be considered.
Applying the same modeling methodology to the one-dimensional Euler equations, a nonlinear model with dimension \( n = 447 \), 67 times smaller than that constructed from the two-dimensional Euler equations, can be built. In the next section, it will be shown that this model provides a turn-around time that allows for efficient tuning of damping parameters to produce a spatially-accurate nonlinear model of the isolator.

### 2.3 One-dimensional Euler equation-based model

A smaller order model of the isolator can be constructed by applying the methodology of Chicatelli and Harley to the one-dimensional Euler equations \[2.88\]

\[
\begin{align*}
\frac{\partial \rho A}{\partial t} + \frac{\partial \rho w A}{\partial z} &= 0 \\
\frac{\partial \rho w A}{\partial t} + \frac{\partial (\rho w^2 + p) A}{\partial z} &= p \frac{\partial A}{\partial z} \\
\frac{\partial \rho EA}{\partial t} + \frac{\partial \rho w HA}{\partial z} &= 0
\end{align*}
\]

Like the model discussed in Section 2.2, the flow variables in this system are density, streamwise velocity, and internal energy, where velocity is a single component in the streamwise direction. Pressure and enthalpy can be calculated from these flow variables: \( p = (\gamma - 1)(E - 0.5w^2) \) and \( H = E + p/\rho \). In the one-dimensional Euler equations, geometry is introduced through terms dependent on the cross-sectional area, \( A \), rather than through spatial partial derivatives. These equations can be further simplified because the cross-sectional area of the isolator is temporally-constant and increases linearly along the \( z \) axis (Figure 2.2).
The cross-sectional area-dependent terms can be separated from the flow variable-dependent terms by expanding the partial derivatives in equation (2.88):

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho w}{\partial z} &= -\frac{1}{A} \left( \rho \frac{\partial A}{\partial t} + \rho w \frac{\partial A}{\partial z} \right) \\
\frac{\partial \rho w}{\partial t} + \frac{\partial \rho w^2 + p}{\partial z} &= -\frac{1}{A} \left( \rho w \frac{\partial A}{\partial t} + \rho w^2 \frac{\partial A}{\partial z} \right) \\
\frac{\partial \rho E}{\partial t} + \frac{\partial w(\rho E + p)}{\partial z} &= -\frac{1}{A} \left( \rho E \frac{\partial A}{\partial t} + w(\rho E + p) \frac{\partial A}{\partial z} \right)
\end{align*}
\] (2.89)

The relationship between the two-dimensional Euler equation-based PBM and the one-dimensional Euler equation-based PBM can be seen through comparison of equations (2.8) and (2.89).

The Euler equations can be expressed in a compact form through definition of the state variable \( u = [\rho \ \rho w \ \rho E]^T \) and the flux vectors \( h(u) \) and \( k(u) \):

\[
h(u) = \begin{bmatrix} \rho w \\ \rho w^2 + p \\ w(\rho E + p) \end{bmatrix} \quad k(u) = -\frac{1}{A} \begin{bmatrix} \rho \frac{\partial A}{\partial t} + \rho w \frac{\partial A}{\partial z} \\ \rho w \frac{\partial A}{\partial t} + \rho w^2 \frac{\partial A}{\partial z} \\ \rho E \frac{\partial A}{\partial t} + w(\rho E + p) \frac{\partial A}{\partial z} \end{bmatrix}
\] (2.90)

Since \( \frac{\partial A}{\partial t} = 0 \) and \( \frac{\partial A}{\partial z} \) is a constant for this isolator, it is possible to further simplify \( k(u) \) by neglecting the derivative with respect to time and by replacing \( \frac{\partial A}{\partial z} \) with the slope of the isolator wall.

Given these definitions, equation (2.89) can be expressed as a single partial differential equation of the form

\[
\frac{\partial u}{\partial t} + \frac{\partial h(u)}{\partial z} = k(u)
\] (2.91)
Application of the chain rule to equation (2.91) yields the term \( \partial h/\partial z = J_h \partial u/\partial z \), where \( J_h \) is the Jacobian of the flux function \( h(u) \):

\[
J_h = \begin{bmatrix}
0 & 1 & 0 \\
-w^2 \frac{\gamma-3}{2} & (3 - \gamma) w & (\gamma - 1) \\
-w(\gamma E - (\gamma - 1)w^2) & \gamma E - \frac{3(\gamma-1)}{2} w^2 & \gamma w
\end{bmatrix}
\] (2.92)

A similar simplification can be made by recognizing \( k(u) \) as a homogeneous function of degree one, \( k(u) = J_k u \), where \( J_k = \partial k/\partial u \):

\[
J_k = -\frac{\partial A}{\partial Z} \begin{bmatrix}
0 & 1 & 0 \\
-w^2 & 2w & 0 \\
-w(\gamma E - (\gamma - 1)w^2) & \gamma E - \frac{3(\gamma-1)}{2} w^2 & \gamma w
\end{bmatrix}
\] (2.93)

These Jacobians are similar to those in equations (2.14) and (2.15), further showing the similarities between the nonlinear models.

Detailed in Appendix B, equation (2.91) can be spatially discretized following the method described in Section 2.2.6, where flux-splitting is implemented to maintain stability. It should be noted that the boundary condition functions have been modified from those defined in Section 2.2.6 to include ‘set-points’ for flow variables when the flow is supersonic:

\[
B_{bc}^{in}(u) = \begin{bmatrix}
\rho - \rho^* \\
(\rho w) - (\rho w)^* \\
(\rho E) - (\rho E)^*
\end{bmatrix}
\]

\[
B_{bc, sup}^{out}(u, p_{in}) = \begin{bmatrix}
\rho - \rho^* \\
p - p_{in} \\
(\rho E) - (\rho E)^*
\end{bmatrix}
\] (2.94)

\[
B_{bc, sub}^{out}(u, p_{in}) = p - p_{in}
\]
The set-point terms contribute to construction of a matrix, $A^*(x)$, in the affine characterization of the nonlinear model.

At a general grid point $k$, the discretized governing equations have the form in equation (2.21), but at the boundaries these equations have forms defined by the boundary condition treatment: equation (2.23) at $k = 1$ (and $k = N_z$ when flow is supersonic at the exit plane) and equation (2.28) at $k = N_z$ (when flow is subsonic). Combining these state equations, as described in Section 2.2.7, leads to construction of an intermediate nonlinear model, with state matrix $\tilde{A}(x)$ and input vector $\tilde{B}(x, p_{in})$; expanding $\tilde{B}(x, p_{in})$ then contributes terms to $A(x)$ and leads to construction of $A^*(x)$ and $B(x)$.

The relationship between the two-dimensional and one-dimensional Euler equations is reflected in the structures of the state and input matrices. Because the one-dimensional Euler equations describe the flow along a single radial plane (that is, for a single index $i$), it is not surprising that the structure of $A(x)$ (equation (B.20)) is the same as the diagonal submatrix $A_i(x)$ (equation (2.38) or equation (2.39)). Similarly, the submatrices $B_{N_z,\text{sub}}(x)$ and $B_{N_z,\text{sup}}(x)$, defined for the one-dimensional Euler equation-based model in equations (B.23) and (B.24), are like $B_{(i,N_z),\text{sub}}(x)$ and $B_{(i,N_z),\text{sup}}(x)$ in equations (2.25) and (2.24). Construction of the state and input matrices can be found in Appendix B, where the matrix $A^*(x)$ is also derived from the boundary condition functions and defined in equation (B.22).

The nonlinear isolator model is completed by defining an output function that is affine with respect to the input. Like the two-dimensional Euler equation-based model, the output is defined as pressure measurement at various locations along the isolator, which are not specified here to maintain generality. Since this is the same output defined for the two-dimensional Euler equation-based model, the same relationship exists between $C(x)$ and $C_i(x)$ as between $A(x)$ and $A_i(x)$. The feedthrough
matrix $D$ is similarly related to $D_i$: a zero matrix with a 1 in the last element, if the output at $k = N_z$ is of interest. Appendix B contains additional detail about the derivation and structure of $C(x)$.

The model can be expressed in the nonlinear, affine form in equation (2.95) given the definitions of each system matrix, derived in the appendix. A damping scheme related to that derived in Section 2.2.9 for the two-dimensional Euler equation-based model will be implemented to allow for shock propagation to be captured by this model.

$$
\dot{x} = A(x)x + B(x)p_{in} + A^*(x)x^* \\
y = C(x)x + Dp_{in}
$$

(2.95)

2.3.1 Damping modification

The damping scheme in equation (2.67) is adapted for the one-dimensional Euler equation-based model by neglecting finite differences taken along the radial direction, resulting in the form

$$(D_z^2 - D_z^4)u = \{ \nabla_z [\lambda k_{+0.5} \epsilon_k^2 \Delta z] - \nabla_z [\lambda k_{+0.5} \epsilon_k^4 \Delta z \nabla \Delta z] \} u_k
$$

(2.96)

where the backwards-difference operator is indicated by $\nabla$ and the forward-difference by $\Delta$. The scaling factors are related to those in equation (2.68):

$$
\lambda_k = \frac{1}{2} [\lambda_{k+1} + \lambda_{k+1}]
\
\epsilon_k^2 = \kappa^2 \max \{ \nu_k, \nu_{k+1} \}
\
\epsilon_k^4 = \max \{ 0, \kappa^4 - \epsilon_k^2 \}
$$

(2.97)

where $\lambda_{k+1}$ are related to the maximum eigenvalues of $J_h$ at grid point $k^*$ ($\lambda_z = |w| + c$) and $\kappa^{2,4}$ are, nominally, 1/4 and 1/256. The most influential term in equation (2.97)
is $\nu_k$, which reads as
\[
\nu_k = \frac{|p_{k+1} - 2p_k + p_{k-1}|}{|p_{k+1} + 2p_k + p_{k-1}|}
\] (2.98)
and is related to the pressure gradient in the flow.

These highly-nonlinear terms can be used to construct damping matrices $P^2(x)$ and $P^4(x)$, which multiply the state vector $x$ in the modified model:
\[
\dot{x} = A(x)x + B(x)u + A^*(x)x^* + P^2(x)x - P^4(x)x
\]
\[
= (A(x) + P^2(x) - P^4(x))x + B(x)u + A^*(x)x^*
\] (2.99)
\[
= \hat{A}(x)x + B(x)u + A^*(x)x^*
\]
The second- and fourth-order damping matrices for this model are related to those constructed in equations (2.77) and (2.87) in the same way the state matrices of the two models are related: $P^{2,4}(x)$ for the one-dimensional Euler equation-based model (equations (B.35) and (B.39)) have the same structure as the diagonal blocks $P^{2,4}_d(x)$ (equations (2.76) and (2.86)). Complete details about the damping scheme, including the boundary condition treatment, can be found in Appendix B.

2.3.2 Model Validation

The model in equation (2.99) was implemented as a SIMULINK® model with separate s-functions written to calculate each matrix. Simulations were run on two different machines: a Dell Inspiron 1501 laptop with 2.62 GB RAM and a 1.8 GHz processor and a Dell Precision T1500 workstation with 16 GB RAM and a quad-core 2.8 GHz processor. On the laptop, simulations could be completed in 1 to 1.5 hours; run-time was decreased by a factor of 4, to nearly 20 minutes, on the workstation. The quick simulation time and the ability to utilize multiple computers improved the ability to evaluate and accurately tune the model.
Along with evaluating the effects of adjusting damping parameters on the model output, the relationship between initial conditions or inputs and the scaling factors were also studied during model validation. The pressure profiles of four initial conditions are shown in Figure 2.28. Two of the initial conditions are steady-state flow conditions: $x_0^1$, the tare condition (no shock wave has formed), and $x_0^4$, the steady-state condition when $p_b = p_bA = 218.577$ kPa. The other initial conditions are non-steady-state conditions that occur during transition between two steady-states: $x_0^2$ represents the condition when the shock has formed just inside the isolator exit plane and $x_0^3$ when the shock has traveled 20 cm upstream from the exit plane. It was found that $x_0^2$ did not yield a steady-state shock profile, so the results presented here are for simulation with the other three initial conditions, with focus mostly on simulations with $x_0^3$ and $x_0^4$. 

Figure 2.28: Pressure and flow variable (density, velocity and internal energy) profiles for the four initial conditions considered during validation. Note that only $x_0^1$ and $x_0^4$ are steady-state flow conditions.
Figure 2.29: Expected pressure and flow variable (density, velocity and internal energy) profiles for each input considered during validation.

Four inputs were also considered: \( p_{in} = p_{BC} = 226.313 \text{ kPa} \), \( p_{in} = p_{BD} = 235.78 \text{ kPa} \), \( p_{in} = 227.528 \text{ kPa} \) and \( p_{in} = 241.318 \text{ kPa} \). The steady-state CFD solutions for each backpressure are shown in Figure 2.29. The model was tuned for input to produce an accurate steady-state shock location, and results suggest that an input-dependent scaling function is necessary to improve the accuracy of the model.

The undamped model (equation (2.95)) was simulated to support the decision to include damping in the model. In addition to this simulation, 44 simulations of the model with damping were run to study the effects of tuning the damping parameters, and the choice of initial condition and input, on the output (pressure profiles). Most of the simulations were run with initial conditions \( x_0^3 \) and \( x_0^4 \) to study the effects of tuning the damping parameters in \( P^2(x) \) and \( P^4(x) \) and the relationship between the input and scaling factors, respectively. Analysis of the results was followed by
construction of an input-dependent scaling function to modify the model structure and improve the accuracy of the model output.

Simulation Results and Analysis

The results presented in this section are representative of the full set of simulation results collected for various combinations of initial conditions, input and scaling factors. For each simulation, a single figure, with separate subplots for the pressure, density, velocity and internal energy profiles, is presented. It is easy to see the propagation of the shock wave from any of these profiles, although the pressure profile is most often used. The profiles of the flow variables can be used to determine if a variable contributes to problems, such as with the undamped model.

Simulation of the Undamped Model. The model in equation (2.95) was simulated with initial conditions $x_0^3$ and an input $p_{bc}$. It can be seen from the results in Figure 2.30 that the shock does not propagate despite the increase in backpressure. It is also observed that the shock profile becomes more sharply defined as the simulation progresses. The flow variable profiles reveal that, in particular, the absence of shock motion causes the internal energy of the system to increase downstream of the shock. Along with this, sharp decreases in density and velocity contribute to the resulting pressure profile.

These changes downstream of the shock maintain the expected value of pressure in this region, but the absence of shock motion is undesirable and suggests the need for damping in the model. When implemented in the model, second-order damping is implemented in the entire flow domain, but fourth-order damping was implemented only in regions of subsonic flow. Before implementing the model, it was modified to explicitly include the tunable scaling factors $\alpha_{sc}$ and $\beta_{sc}$. These are found by assuming the scaling on each $\alpha_{k+1}$ and $\beta_{k+1}$ are the same for each $k$; the structure of
Profiles of Pressure (T) and Flow Variables $\rho$, w, E (B) (no damping)

Figure 2.30: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the undamped model, where the shock does not propagate upstream, despite a rise in backpressure.

each matrix makes it possible to factor out these scaling factors so that the damping terms become $\alpha_{sc} P^2(x)$ and $\beta_{sc} P^4(x)$, where $P^2(x)$ and $P^4(x)$ are the matrices in equations (B.35) and (B.39), respectively. This modification allows each simulation to be identified by the scaling factors $\alpha_{sc}$ and $\beta_{sc}$, the initial conditions and the input.

Simulation with initial condition $x^1_0$. The model with the tare initial condition $x^1_0$ was simulated with input $p_{bc}$ to investigate if a shock wave would form and propagate to a steady-state location in the isolator (for an appropriate choice of $\alpha_{sc}$). For scaling factors $\alpha_{sc} = 2.9 \cdot 10^4$ and $\beta_{sc} = 2 \cdot 10^4$, shock formation and propagation can be seen in Figure 2.31. For this simulation, fourth-order damping was necessary because of the large magnitude of $\alpha_{sc}$: for $\alpha_{sc} > 10^4$, simulations terminated due to numerical errors unless an appropriate $\beta_{sc} > 0$ was chosen. It should also be noted that the shock profile was smeared across a 20 cm region of the isolator due to the large magnitude of the scaling factors necessary for this simulation.
Figure 2.31: Pressure and flow variable (density, velocity and internal energy) profiles for simulation of the model with scaling factors $\alpha_{sc} = 2.9 \times 10^4$ and $\beta_{sc} = 2 \times 10^4$, and an initial condition at tare. A shock wave can be seen to form in the isolator and propagate to the expected steady-state location.

For these scaling factors, the shock wave establishes a steady-state near $z = 67$ cm, the expected location when $p_{in} = p_{bc}$. The simulation was temporally accurate in comparison to the CFD data, as steady-state was established after 25 ms. Although transient CFD data was not available for this particular case, for other simulations steady-state conditions were established after 25 to 30 ms. Once in its steady-state location, the shock wave oscillated about this location for the remainder of the simulation, requiring a small time-step to capture this motion, slowing the progress so that it took nearly 2 hours to complete the simulation.

Simulation with Initial Condition $x_0^3$. The effects of scaling the second- and fourth-order damping matrices through the choices of $\alpha_{sc}$ and $\beta_{sc}$ were studied through simulation of the model with the non-steady-state initial condition $x_0^3$. It was found that only the second-order damping term was necessary for shock propagation to be
observed and that $\alpha_{sc}$ affected the steady-state location of the shock wave while $\beta_{sc}$ most affected the profile shape. The four results shown here are representative of the twenty-three simulations run with this initial condition.

*Simulation with Second-Order Damping.* The scaling factor $\alpha_{sc}$ must be above a certain value for shock propagation to occur and, as $\alpha_{sc}$ is further increased above this minimum value, the location of the steady-state shock wave moves further upstream. It was also found that, for $\alpha_{sc}$ above a certain value (in this case, $\alpha_{sc} = 1.01 \cdot 10^4$), fourth-order damping was necessary to avoid numerical errors during simulation, as was observed during simulation of the model with initial condition $x_{10}^1$.

Simulation results with $\alpha_{sc} = 500$ are shown in Figure 2.32, where the shock can be seen to propagate upstream, establishing a steady-state at around $z = 55$ cm, 12 cm upstream of the expected location. The density and velocity profiles are smoother than those of the internal energy, where some oscillation is observed downstream of the shock. Although this behavior is not seen in the CFD results, it appears compensated for by the other flow variables, since the pressure profiles are smooth. In addition to the spatial, the model is temporally inaccurate as it takes 50 ms for the steady-state to establish, 20 ms longer than in the CFD simulation.

Decreasing the scaling factor to $\alpha_{sc} = 400$ produced the most accurate steady-state shock results for $p_{in} = p_{bc}$, shown in Figure 2.33. The results are generally similar to those in Figure 2.32, with the exception of the final location of the shock and some reduction of oscillations in the internal energy profiles.

This simulation was run for 0.1 sec of data to verify how long it took for the flow field to establish a steady-state. Because of the similarity to Figure 2.33, results are not presented here, but it was found that it took between 50 and 55 ms for the shock to settle to a steady-state location, the same temporal accuracy as found for
Figure 2.32: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the system with $\alpha_{sc} = 500$. The inclusion of damping in the model allows for the shock wave to propagate to 55 cm along the streamwise direction.

Simulation with $\alpha_{sc} = 500$. Additional modification to the model may be necessary in order to achieve temporal accuracy.

The effect on the simulation results of scaling by the time-step, $(\Delta T)^{-1}$ in $A(x)$ and $B(x)$, introduced through the boundary condition treatment, was then investigated. In general, $\Delta T = 10^{-4}$ s, but simulations were run with $\Delta T = 0.01$ sec and $\Delta T = 10^{-8}$ sec to compare the results. It was found that these adjustments did not affect the spatial or temporal accuracy, but did affect the time necessary to complete the simulation: it took nearly 2 days to collect that data when $\Delta T = 10^{-8}$ sec, compared to 20 min when $\Delta T = 10^{-4}$ sec. When $\Delta T = 0.01$ sec, the effects were not as noticeable, but the original choice ($\Delta T = 10^{-4}$ sec) was retained for consistency with the CFD data.

*Simulation with Second- and Fourth-Order Damping.* Fixing $\alpha_{sc} = 400$, which yields spatially accurate steady-state results for $p_{in} = p_{bc}$, the effects of tuning
Figure 2.33: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the model with $\alpha_{sc} = 400$. This scaling factor allows the shock to propagate to 68 cm, as expected from CFD data.

$\beta_{sc}$ were investigated. The results in Figures 2.34 and 2.35, where $\beta_{sc} = 10^4$ and $\beta_{sc} = 1000$, respectively, can be compared to those in Figure 2.33, where $\beta_{sc} = 0$, to how fourth-order damping affects the results. (These effects are most clear through comparison of the internal energy profiles.) Although it was found that implementation of fourth-order damping in the entire flow domain had little effect on the results, fourth-order damping was implemented only in regions of subsonic flow, as suggested by Pulliam [41]. With only second-order damping, oscillation in the energy profile (in the region downstream of the shock) peaks around 5 ms and dies out around 30 ms. The introduction of fourth-order damping, with $\beta_{sc} = 10^4$, causes the oscillations to become more pronounced, and to die out after 45 ms. In addition, small oscillations are introduced near the exit plane due to this additional damping. When reduced to $\beta_{sc} = 1000$, these small oscillations remain, but the larger oscillations are less-pronounced, as shown in Figure 2.35.
Figure 2.34: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the model with $\alpha_{sc} = 400$ and $\beta_{sc} = 10^4$. Fourth-order damping appears to only effect the shape of the profiles downstream of the shock.

Figure 2.35: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the model with $\alpha_{sc} = 400$ and $\beta_{sc} = 1000$. 
Table 2.5: Scaling factors $\alpha_{sc}$ necessary to achieve accurate steady-state shock location for simulation with initial condition $x_0^4$ and each $p_b$ under study.

<table>
<thead>
<tr>
<th>$p_b$</th>
<th>$\alpha_{sc}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{bC} = 226.313$ kPa</td>
<td>360</td>
</tr>
<tr>
<td>$p_{bD} = 235.78$ kPa</td>
<td>280</td>
</tr>
<tr>
<td>33 psi = 227.528 kPa</td>
<td>337</td>
</tr>
<tr>
<td>35 psi = 241.318 kPa</td>
<td>272</td>
</tr>
</tbody>
</table>

Although these results suggest that it may be possible to simplify the model by implementing only second-order damping, additional simulations have shown that large $\alpha_{sc}$ are necessary to achieve accuracy in steady-state and $\beta_{sc} \neq 0$ is necessary to avoid numerical errors in these simulations. A fixed value (e.g. $\beta_{sc} = 10^4$) may be chosen since tuning $\beta_{sc}$ does not affect spatial accuracy of the profiles.

**Simulation with Initial Condition $x_0^4$ and Different Inputs.** The model was simulated with the steady-state initial condition $x_0^4$ and the four inputs: $p_{in} = p_{bC} = 226.313$ kPa, $p_{in} = p_{bD} = 235.78$ kPa, $p_{in} = 227.528$ kPa and $p_{in} = 241.318$ kPa to investigate how sensitive $\alpha_{sc}$ is to changes in $p_{in}$. Figures 2.36 and 2.37 are representative of the eighteen simulations that were run for these initial condition and input pairs, where it was necessary to tune $\alpha_{sc}$ for each input to achieve accurate steady-state profiles. These scaling factors are summarized in Table 2.5.

The results in Figure 2.36 are from a simulation where $p_{in} = p_{bC}$ and $\alpha_{sc}$ is tuned appropriately (in this case, $\alpha_{sc} = 360$). When the input is changed to $p_{in} = p_{bD}$, the profiles in Figure 2.37 result, where the steady-state shock wave is established 12 cm upstream of the expected location (13.2% of the isolator length $L = 90.678$ cm). These results suggest a dependency between the scaling factor necessary to obtain spatially-accurate simulation results and the input $p_{in}$, where the shock wave location error increases as the input is increased further from that for which $\alpha_{sc}$ was tuned. Similar results were found from simulation of other ‘mismatched’ pairs of $p_{in}$ and $\alpha_{sc}$. 

114
Figure 2.36: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the model with initial condition $x_0^4$, $\alpha_{sc} = 360$ and $p_b = p_D$. The shock establishes a location around 66 cm, as expected from CFD data.

Figure 2.37: Pressure and flow variable (density, velocity, and internal energy) profiles from simulation of the model with initial condition $x_0^4$, $\alpha_{sc} = 360$ and $p_b = p_D$. This $\alpha_{sc}$ yields a steady-state shock location around 12 cm upstream from the expected location for this input, suggesting $\alpha_{sc}$ depends on $p_b$. 

115
Additional simulations were run for each input with the scaling factors $\alpha_{sc} = \{280 \, , \, 290, \, ..., \, 330, \, 350\}$ and shock locations are calculated as the location at which pressure increased above 10% of the tare value. In Figure 2.38, the steady-state shock location as a function of $\alpha_{sc}$ is shown in the left subplot, where the dashed line represents the initial shock position ($x_0^4$) and the dotted lines the expected location (determined from CFD), the middle plot shows the shock location errors, and the right plot shows the relative error calculated with respect to $L$.

These results suggest that an input-dependent scaling factor $\alpha_{sc}(p_{in})$, constructed by curve-fitting a set of $(p_{in}, \alpha_{sc})$ pairs to find a continuous scaling function, may be introduced to the model to improve steady-state accuracy:

\[
\dot{x} = (A(x) + \alpha_{sc}(p_{in})P^2(x) - \beta_{sc}P^4(x))x + B(x)p_{in} \\
y = C(x)x + Dp_{in}
\]  
(2.100)
This modification is discussed in the next section.

**Constructing an input-dependent scaling function**

The five sets of steady-state CFD data considered during validation of the one-dimensional Euler equation-based PBM (shown in Figure 2.29) were augmented with four additional sets of steady-state data provided by AFRL. These data sets provided nine initial and final conditions for which the nonlinear model could be tuned to find scaling factors for which the steady-state shock location was accurate. In total, sixty-four pairs of scaling factors were found, and the input-dependent scaling functions were \( \alpha_{sc}(p_{in}) \) was found by performing a curve fit on sets of \( \alpha_{sc} \).

The values of backpressure at which steady-state CFD data was available are in Table 2.6, where it should be noted that the data sets in Figure 2.29 (those with backpressure \( p_b = 218.577, 226.313, 235.78, 227.528, \) and \( 241.318 \) kPa) were collected from CFD simulations where the transient data was of interest, while only the steady-state solutions were of interest for the other simulations. This is hypothesized to contribute to the discrepancies in Table 2.6, where the shock appears to move upstream (downstream) for some backpressure decreases (increases), such as between 213.737 kPa and 218.577 kPa.

The nonlinear model is tuned for each pair of initial conditions \( x_0 \) and input \( p_{in} \) defined from the values in Table 2.6. For simulation of the model with an increase in backpressure, \( \alpha_{sc} \) was adjusted while \( \beta_{sc} = 0 \). For a backpressure decrease, it was necessary for \( \beta_{sc} \neq 0 \) because of the increase in \( \alpha_{sc} \) to achieve the correct shock motion; for simplicity, the factor was fixed at \( \beta_{sc} = 50000 \). Table 2.7 shows the scaling factors \( \alpha_{sc} \) for each simulation that was run; the eight simulations that showed the discrepancy between the direction of shock propagation and input change were not tuned as the scaling factors would not be consistent with those of similar cases. For
Table 2.6: Backpressures $p_b$ for which steady-state CFD data was available, and the steady-state shock location for each input.

<table>
<thead>
<tr>
<th>$p_b$ (psi)</th>
<th>$p_b$ (kPa)</th>
<th>$z_{sh,ss}$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.5</td>
<td>210.29</td>
<td>88.879</td>
</tr>
<tr>
<td>31</td>
<td>213.737</td>
<td>79.637</td>
</tr>
<tr>
<td>31.702</td>
<td>218.577</td>
<td>85.065</td>
</tr>
<tr>
<td>32</td>
<td>220.632</td>
<td>61.76</td>
</tr>
<tr>
<td>32.824</td>
<td>226.313</td>
<td>66.369</td>
</tr>
<tr>
<td>33</td>
<td>227.528</td>
<td>64.07</td>
</tr>
<tr>
<td>33.5</td>
<td>230.974</td>
<td>36.383</td>
</tr>
<tr>
<td>34.197</td>
<td>235.78</td>
<td>42.774</td>
</tr>
<tr>
<td>35</td>
<td>241.318</td>
<td>31.033</td>
</tr>
</tbody>
</table>

For example, a backpressure increase from 213.737 kPa to 218.577 kPa (31 psi to 31.702 psi) suggests that $\alpha_{sc}$ should be on the order of 100 and $\beta_{sc} = 0$, but the expected direction of shock propagation suggests instead that $\alpha_{sc}$ be on the order of $10^5$ and $\beta_{sc} = 50000$. 
Table 2.7: Scaling factors $\alpha_{sc}$ necessary to achieve the most accurate steady-state shock location for simulation of the nonlinear model with each $(x_0, p_{in})$ pair. Note that the initial condition $x_0$ is the steady-state profile for the backpressure $p_{b0}$.

<table>
<thead>
<tr>
<th>$p_{b0}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td>515</td>
<td>366.5</td>
<td>355</td>
<td>265</td>
<td>255</td>
<td>290</td>
<td>212</td>
<td>215</td>
</tr>
<tr>
<td>213.737</td>
<td>139000</td>
<td></td>
<td></td>
<td>355</td>
<td>250</td>
<td>240</td>
<td>290</td>
<td>210</td>
<td>215</td>
</tr>
<tr>
<td>218.577</td>
<td>145000</td>
<td></td>
<td></td>
<td>770</td>
<td>360</td>
<td>337</td>
<td>460</td>
<td>280</td>
<td>272</td>
</tr>
<tr>
<td>220.632</td>
<td>139000</td>
<td>135000</td>
<td>130000</td>
<td></td>
<td></td>
<td>290</td>
<td>214</td>
<td>215</td>
<td></td>
</tr>
<tr>
<td>226.313</td>
<td>135000</td>
<td>130000</td>
<td>180000</td>
<td></td>
<td></td>
<td>340</td>
<td>460</td>
<td>280</td>
<td>272</td>
</tr>
<tr>
<td>227.528</td>
<td>135000</td>
<td>130000</td>
<td>170000</td>
<td></td>
<td></td>
<td>177000</td>
<td></td>
<td>540</td>
<td>310</td>
</tr>
<tr>
<td>230.974</td>
<td>135000</td>
<td>125000</td>
<td>145000</td>
<td>140000</td>
<td>140000</td>
<td>135000</td>
<td></td>
<td></td>
<td>215</td>
</tr>
<tr>
<td>235.78</td>
<td>135000</td>
<td>130000</td>
<td>170000</td>
<td>125000</td>
<td>135000</td>
<td>120000</td>
<td></td>
<td></td>
<td>272</td>
</tr>
<tr>
<td>241.318</td>
<td>135000</td>
<td>130000</td>
<td>150000</td>
<td>125000</td>
<td>130000</td>
<td>131000</td>
<td>110000</td>
<td>110000</td>
<td></td>
</tr>
</tbody>
</table>
It was generally possible to tune $\alpha_{sc}$ so that the shock location error was within 1 cm for a backpressure increase, and within 5 cm for a backpressure decrease, which is less than the one duct height that the shock is expected to oscillate in steady-state. (This loss of accuracy is related to the smearing of the pressure profile related to the large magnitudes of $\alpha_{sc}$ and $\beta_{sc}$.) However, in some cases numerical errors were encountered when tuning $\alpha_{sc}$, so little more than 1 or 2 cm of shock propagation was possible because it was not possible to adjust $\alpha_{sc}$ to a large enough value. In order to more easily analyze the steady-state shock location errors, Table 2.8 contains errors that have been normalized by the duct height $D_H$, which is a linear function of $z$ and has an average value $\bar{D}_H = 11.5$ cm, where errors under one duct height are acceptable as the shock wave oscillates up to one duct height in steady-state.
Table 2.8: Relative shock location errors, calculated with respect to the isolator duct height $D_H = 2R(z)$, where $R(z)$ is the radius of the isolator, for simulations with the tuning parameters in Table 2.7.

<table>
<thead>
<tr>
<th>$p_\omega$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>$p_{in}$ (kPa)</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td>0</td>
<td>-0.8387</td>
<td>0</td>
<td>-0.4694</td>
<td>-0.5396</td>
<td>-0.0611</td>
<td>0</td>
<td>-0.0575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>213.737</td>
<td>-1.2956</td>
<td></td>
<td>-0.0673</td>
<td>-0.0665</td>
<td>-0.067</td>
<td>-0.0611</td>
<td>0.0646</td>
<td>-0.0575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>218.577</td>
<td>-0.2708</td>
<td></td>
<td>-0.0673</td>
<td>-0.0665</td>
<td>0.0668</td>
<td>-0.0611</td>
<td>0.0646</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>220.632</td>
<td>-0.2154</td>
<td>0</td>
<td>-2.5826</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.0611</td>
<td>-0.0644</td>
<td>-0.0575</td>
</tr>
<tr>
<td>226.313</td>
<td>-0.2708</td>
<td>0.1195</td>
<td>-2.1007</td>
<td></td>
<td>-0.067</td>
<td>-0.0611</td>
<td>0.0646</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>227.528</td>
<td>-0.2708</td>
<td>0.2948</td>
<td>-1.692</td>
<td>-0.4694</td>
<td>-0.1219</td>
<td>-0.0644</td>
<td>0.058</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>230.974</td>
<td>-1.2956</td>
<td>-1.0193</td>
<td>-2.0321</td>
<td>-0.3376</td>
<td>-2.0217</td>
<td>-1.8842</td>
<td></td>
<td>-0.0575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>235.78</td>
<td>-0.2708</td>
<td>0.1195</td>
<td>-1.1602</td>
<td>0.4005</td>
<td>-0.3344</td>
<td>-0.2014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>241.318</td>
<td>-0.2708</td>
<td>0.2948</td>
<td>-1.6246</td>
<td>0.5977</td>
<td>0.0662</td>
<td>-0.067</td>
<td>0.7592</td>
<td>-0.1923</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Grouping the scaling factors in Table 2.7 first by the initial condition for the simulation for which they were found and then by whether they were found for an increase or decrease it backpressure, two scaling functions $\alpha_{sc}^{\pm}(p_{in})$ can be constructed for each initial condition (when at least two simulations could be run). The scaling factors were found to decrease nonlinearly as input increased, so higher order polynomials of $p_{in}$ were considered for $\alpha_{sc}^{\pm}(p_{in})$. The trend of these scaling factors, over the pressure range of interest, suggested that these functions be even-order polynomials, so order 2, 4, 6, and 8 polynomials were constructed for $\alpha_{sc}^{\pm}(p_{in})$, where the quartic functions produced fits with the most accurate estimates of $\alpha_{sc}$ for the largest number of pairs of initial conditions and input. The curve fits were constructed by considering the polynomials in the form:

$$\alpha_{sc}(p_{in}) = c_0 + c_1 p_{in} + c_2 p_{in}^2 + \ldots$$

$$= \sum_{i=0}^{\infty} c_i p_{in}^i$$

where the coefficients $c_i$ are to be found. The powers of $p_{in}$ were collected in a row-vector, $P_{in,i}$, the unknown coefficients in the column-vector, $C$, and, for a given back-pressure, $A_{sc,i} = P_{in,i} C$, where $A_{sc,i}$ is the scaling factor $\alpha_{sc}$. Stacking these expression for each initial condition produces the equation $A_{sc} = P_{in} C$. Pre-multiplication by the pseudoinverse of $P_{in}$ solves the system, and yielding the matrix of unknown coefficients, $C$; in MATLAB®, the Moore-Penrose pseudoinverse $P_{in}^+$ is calculated, so that $C = P_{in}^+ A_{sc}$.

In addition to the fits constructed for each initial condition, two other fits were performed: one with the pressure in psi rather than Pascals and the other considering the scaling factors in Table 2.7 without differentiation by initial condition. The fits produced when pressure is expressed in Pascals were the most accurate, so these were
considered together with the fits constructed from the full set of data in validating the modified model. The justification for the latter fit will be discussed subsequently.

Instead of considering every fit constructed for the data in Table 2.7, three fits, each, were considered for $\alpha_{sc}^+(p_{in})$ and $\alpha_{sc}^-(p_{in})$. Recalling the discrepancy between the CFD data sets considered in validation of the PBM (Figure 2.29) and those collected for constructing the scaling function, minor dependence on the initial conditions can be seen. Grouping the CFD data sets by the use for which they were obtained, it can be seen that $\alpha_{sc}$ only depends on the input within the same group of data; e.g. $\alpha_{sc}$ for simulations with $p_{b0} = 210.29$ kPa (30.5 psi) are nearly the same as for simulations with $p_{b0} = 213.737$, 220.632, and 230.974 kPa (31, 32, and 33.5 psi). This suggests that a single fit for each group of data may be considered; since the accuracy of the fits is improved when more data is considered, $\alpha_{sc}^+(p_{in})$ was constructed from data with $p_{b0} = 210.29$ kPa and $p_{b0} = 218.577$ kPa and $\alpha_{sc}^-(p_{in})$ from data with $p_{b0} = 230.974$ kPa and $p_{b0} = 241.318$ Pa. These fits were performed for simulations with the largest number of $\alpha_{sc}$ for a given direction of backpressure change. The coefficients $c_i$, $i \in [1, 5]$, for these fits are given in Table 2.9 (fits 1 and 2, respectively). Fit 3 in the table was constructed from the complete set of scaling functions for an increase in backpressure and for a decrease in backpressure. Had the discrepancy between data sets not existed, this fit would be expected to be similar to the other two fits.

Graphical evaluation of each fit for $\alpha_{sc}^+(p_{in})$ and $\alpha_{sc}^-(p_{in})$ can be done by looking at Figures 2.39 and 2.40, respectively, where the scaling factors in Table 2.7 are shown as scatter plots. The discrepancy in the CFD data is readily seen in Figure 2.39: data shown by the diamond, up-triangle, left-triangle, and two stars represent one group of CFD data, and have scaling factors distinctly different from the other four sets of scaling factors. Fits 1 and 2 are accurate in comparison to the data for which
Table 2.9: Coefficients for the fits of $\alpha_{sc}(p_{in}) = c_0 + c_1 p_{in} + c_2 p_{in}^2 + c_3 p_{in}^3 + c_4 p_{in}^4$ used in validation of the (modified) nonlinear model in equation (2.100).

<table>
<thead>
<tr>
<th></th>
<th>$\alpha_{sc}^+(p_{in})$</th>
<th>$\alpha_{sc}^-(p_{in})$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fit 1 ($p_{b0} = 210.29$ kPa)</strong></td>
<td>$7.43 \times 10^{-17}$</td>
<td>$2.25 \times 10^{-16}$</td>
</tr>
<tr>
<td>$c_0$</td>
<td>$-7.29 \times 10^{-11}$</td>
<td>$-4.31 \times 10^{-10}$</td>
</tr>
<tr>
<td>$c_1$</td>
<td>$6.40 \times 10^{-7}$</td>
<td>$1.99 \times 10^{-6}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$-5.33 \times 10^{-12}$</td>
<td>$-1.67 \times 10^{-11}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$1.12 \times 10^{-17}$</td>
<td>$3.52 \times 10^{-17}$</td>
</tr>
<tr>
<td><strong>Fit 2 ($p_{b0} = 218.577$ kPa)</strong></td>
<td>$2.25 \times 10^{-16}$</td>
<td>$-3.74 \times 10^{-11}$</td>
</tr>
<tr>
<td>$c_0$</td>
<td>$-4.31 \times 10^{-10}$</td>
<td>$3.35 \times 10^{-7}$</td>
</tr>
<tr>
<td>$c_1$</td>
<td>$1.99 \times 10^{-6}$</td>
<td>$3.35 \times 10^{-7}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$-1.67 \times 10^{-11}$</td>
<td>$2.63 \times 10^{-12}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$3.52 \times 10^{-17}$</td>
<td>$5.23 \times 10^{-18}$</td>
</tr>
<tr>
<td><strong>Fit 3 (all data)</strong></td>
<td>$3.85 \times 10^{-17}$</td>
<td>$3.85 \times 10^{-17}$</td>
</tr>
<tr>
<td>$c_0$</td>
<td>$-3.74 \times 10^{-11}$</td>
<td>$-3.74 \times 10^{-11}$</td>
</tr>
<tr>
<td>$c_1$</td>
<td>$3.35 \times 10^{-7}$</td>
<td>$3.35 \times 10^{-7}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$2.63 \times 10^{-12}$</td>
<td>$2.63 \times 10^{-12}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$5.23 \times 10^{-18}$</td>
<td>$5.23 \times 10^{-18}$</td>
</tr>
</tbody>
</table>

they were constructed, where the RMS error of fit 1 is 24.666, suggesting it is slightly more accurate than fit 2 (RMS error of 81.169). Since fit 3 lies in between the other two fits, it is not as accurate, with an RMS error of 180.95, seven times worse than fit 1 and almost twice as worse as fit 2. The difference between the scaling factors found for each group is less clear from results for $\alpha_{sc}(p_{in})$. This is due, in part, to the inability to accurately tune $\alpha_{sc}$ for several simulations when backpressure decreases. Unlike the fits for $\alpha_{sc}^+(p_{in})$, fit 3 does not split the other fits due to outlying $\alpha_{sc}$ that affect the shape of the fit. This is expected to have an effect on the accuracy of the simulation by introducing numerical errors like those encountered when tuning $\alpha_{sc}$.

The nonlinear model is modified from the proposed form in equation (2.100) to explicitly include the two scaling functions $\alpha_{sc}^+(p_{in})$ and the corresponding values of
Figure 2.39: Comparison of the three fits considered for validation. Fits 1 and 2 are for simulations with backpressure increasing from $p_{b0} = 210.29$ kPa (30.5 psi) and $p_{b0} = 218.577$ kPa (31.702 psi), respectively, and fit 3 is calculated for all data.

Figure 2.40: Comparison of the three fits considered for validation. Fits 1 and 2 are for simulations with backpressure decreasing from $p_{b0} = 230.974$ kPa (33.5 psi) and $p_{b0} = 241.318$ kPa (35 psi), respectively, and fit 3 is calculated for all data.
\[ \beta_{sc}: \]

\[ \dot{x} = (A(x) + \alpha_{sc}(p_{in})P^2(x) + \beta_{sc}P^4(x))x + B(x)p_{in} \]

\[ y = C(x)x + Dp_{in} \]

where:

\[ \alpha_{sc}(p_{in}) = \begin{cases} 
\alpha_{sc}^+(p_{in}) & \text{for } p_{in}(k) > p_{in}(k^-) \\
\alpha_{sc}^-(p_{in}) & \text{for } p_{in}(k) < p_{in}(k^-) 
\end{cases} \quad (2.101) \]

\[ \beta_{sc} = \begin{cases} 
0 & \text{for } p_{in}(k) > p_{in}(k^-) \\
50000 & \text{for } p_{in}(k) < p_{in}(k^-) 
\end{cases} \]

The complete characterization of the model in equation (2.101) was implemented by using a trigger to detect a transition in \( p_{in} \); the value of the input at the previous transition (time \( k^- \)) is compared to the new value (at time \( k \)) to determine if the input increased or decreased so the proper scaling function is used. This trigger is necessary if \( p_{in} \) is a single- or multi-step function with multiple time-steps between a change in value.

The model is validated by simulating equation (2.101) with each initial condition and input given in Table 2.6 and each fit given in Table 2.9. For simulations where backpressure increased, results could always be obtained, but large errors in \( \alpha_{sc}^-(p_{in}) \) caused simulation to terminate due to numerical errors related to overestimation of \( \alpha_{sc} \). These are the same errors encountered when tuning \( \alpha_{sc} \), which resulted in steady-state shock location error due to the sensitivity of the model. It would likely be possible to tune \( \beta_{sc} \) to avoid termination of the simulation, but the numerical errors are not predictable and it is not always clear how much \( \beta_{sc} \) would need to be adjusted to fix the problem. In addition, the model output does not demonstrate noticeable changes in time-steps leading up to termination, suggesting the problem may be rectified by adjusting the time step or accuracy of the simulation; in the interest of consistency, these solutions were not implemented in this investigation.
Table 2.10: Estimated values of $\alpha_{sc}$ for each backpressure in Table 2.6, calculated using each of the fits in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{b0}$ (kPa)</th>
<th>fit 1 $\alpha_{sc}^-$</th>
<th>fit 1 $\alpha_{sc}^+$</th>
<th>fit 2 $\alpha_{sc}^-$</th>
<th>fit 2 $\alpha_{sc}^+$</th>
<th>fit 3 $\alpha_{sc}^-$</th>
<th>fit 3 $\alpha_{sc}^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td>130030</td>
<td>564.9</td>
<td>134270</td>
<td>1357.8</td>
<td>134140</td>
<td>560.35</td>
</tr>
<tr>
<td>213.737</td>
<td>135020</td>
<td>490</td>
<td>135830</td>
<td>1120.2</td>
<td>140390</td>
<td>513.23</td>
</tr>
<tr>
<td>218.577</td>
<td>139280</td>
<td>396.19</td>
<td>135590</td>
<td>823.49</td>
<td>144490</td>
<td>450.21</td>
</tr>
<tr>
<td>220.632</td>
<td>140030</td>
<td>361.1</td>
<td>134540</td>
<td>712.81</td>
<td>144410</td>
<td>424.89</td>
</tr>
<tr>
<td>226.313</td>
<td>138370</td>
<td>281.17</td>
<td>128440</td>
<td>462</td>
<td>137870</td>
<td>360.44</td>
</tr>
<tr>
<td>227.528</td>
<td>137260</td>
<td>267.69</td>
<td>126480</td>
<td>420.01</td>
<td>135180</td>
<td>347.86</td>
</tr>
<tr>
<td>230.974</td>
<td>132520</td>
<td>237.18</td>
<td>119550</td>
<td>325.73</td>
<td>124870</td>
<td>314.86</td>
</tr>
<tr>
<td>235.75</td>
<td>121730</td>
<td>215.55</td>
<td>106330</td>
<td>261.43</td>
<td>103370</td>
<td>276.24</td>
</tr>
<tr>
<td>241.318</td>
<td>102630</td>
<td>224.88</td>
<td>85420</td>
<td>297.22</td>
<td>67430</td>
<td>244.214</td>
</tr>
</tbody>
</table>

The relative steady-state shock location errors and errors in $\alpha_{sc}^\pm(p_{in})$ were used to evaluate the simulation results. Due to inaccuracy in the fits and the high sensitivity to changes in $\alpha_{sc}$, it was not expected that these results would be near the values in Table 2.8, but the most accurate fit can be determined by comparing the results from simulations with each fit.

The relative steady-state shock location errors (with respect to $D_H$) are shown in Tables 2.11, 2.13, and 2.15 for each fit, respectively, and the relative errors of the scaling factors in Table 2.10 with respect to those in Table 2.7 are shown in Tables 2.12, 2.14, and 2.16. Note that a negative relative shock location error implies that the shock has propagated further downstream than expected and when the relative error in $\alpha_{sc}$ is negative, the scaling factor is less than expected.

Tables 2.11 and 2.12 suggest that fit 1 produces the most accurate results for simulation with initial conditions $p_{b0} = 210.29, 213.737, 220.632$ and $230.974$ kPa (30.5, 31, 32 and 33.5 psi). When implemented with an initial condition from the other group of data, it was typical for little shock propagation to be seen when backpressure increased since $\alpha_{sc}^\pm(p_{in})$ underestimated the expected scaling factor. There are 13 cases
when the shock does not propagate, which are related to the cases where $\alpha_{sc}(p_{in})$ is the most inaccurate. 10 of the 36 simulations with a decrease of backpressure terminated due to numerical errors caused by overestimation of $\alpha_{sc}$. For two of these cases, backpressure decreasing from $p_{b0} = 230.974$ kPa (33.5 psi) to $p_{in} = 226.313$ kPa (32.824 psi) and to $p_{in} = 227.528$ kPa (33 psi), this error was encountered for simulation with each fit.
Table 2.11: Shock location errors relative to the isolator duct height $D_H = R(z)$, where $R(z)$ is the radius of the isolator, for simulation of the nonlinear model in equation (2.101) with $\alpha_{sc}(p_{in})$ from fit 1, with coefficients in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{in}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td>0.7371</td>
<td>-1.692</td>
<td>-0.1347</td>
<td>-1.1486</td>
<td>-1.0824</td>
<td>1.0862</td>
<td>-0.1285</td>
<td>-0.283</td>
</tr>
<tr>
<td>213.737</td>
<td>-1.6218</td>
<td></td>
<td>-1.7597</td>
<td>-0.2023</td>
<td>-1.2165</td>
<td>-1.15</td>
<td>1.0204</td>
<td>-0.1285</td>
<td>-0.283</td>
</tr>
<tr>
<td>218.577</td>
<td>-0.2154</td>
<td>-0.6256</td>
<td></td>
<td>1.9413</td>
<td>1.5463</td>
<td>1.7433</td>
<td>4.1167</td>
<td>3.5688</td>
<td>0.975</td>
</tr>
<tr>
<td>220.632</td>
<td>-0.3267</td>
<td>-1.7638</td>
<td></td>
<td></td>
<td>-1.2165</td>
<td>-1.15</td>
<td>1.0204</td>
<td>-0.1285</td>
<td>-0.283</td>
</tr>
<tr>
<td>226.313</td>
<td>-0.3833</td>
<td></td>
<td>-2.1007</td>
<td>-0.6764</td>
<td></td>
<td>0.1334</td>
<td>2.5406</td>
<td>1.9849</td>
<td>0.975</td>
</tr>
<tr>
<td>227.528</td>
<td>-0.2708</td>
<td>-1.4907</td>
<td>-1.6246</td>
<td>-0.1347</td>
<td>-1.2165</td>
<td></td>
<td>2.4101</td>
<td>1.8537</td>
<td>1.4265</td>
</tr>
<tr>
<td>230.974</td>
<td>-2.7913</td>
<td></td>
<td>-1.7451</td>
<td></td>
<td></td>
<td></td>
<td>-0.5065</td>
<td>-0.283</td>
<td></td>
</tr>
<tr>
<td>235.78</td>
<td>-0.6154</td>
<td>-1.8323</td>
<td>-2.9959</td>
<td>-0.3376</td>
<td></td>
<td></td>
<td></td>
<td>0.975</td>
<td></td>
</tr>
<tr>
<td>241.318</td>
<td>-1.4907</td>
<td>-2.7205</td>
<td>-1.0817</td>
<td>-2.6662</td>
<td>-1.5533</td>
<td>0.0615</td>
<td></td>
<td>-1.2777</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.12: Relative errors in estimated $\alpha_{sc}$, calculated with respect to the expected values in Table 2.7, for simulation of the model with $\alpha_{sc}(p_{in})$ defined from the coefficients for fit 1 in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{in}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>213.737</td>
<td>-0.0646</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>218.577</td>
<td>-0.1033</td>
<td>-0.531</td>
<td>-0.219</td>
<td>-0.2057</td>
<td>-0.4844</td>
<td>-0.2302</td>
<td>-0.1732</td>
<td></td>
<td></td>
</tr>
<tr>
<td>220.632</td>
<td>-0.0646</td>
<td>0.0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.1821</td>
<td>0.0073</td>
<td>0.046</td>
</tr>
<tr>
<td>226.313</td>
<td>-0.0368</td>
<td>-0.2262</td>
<td></td>
<td></td>
<td></td>
<td>-0.2127</td>
<td>-0.4844</td>
<td>-0.2302</td>
<td>-0.1732</td>
</tr>
<tr>
<td>227.528</td>
<td>-0.0368</td>
<td>0.0386</td>
<td>-0.1807</td>
<td>-0.2183</td>
<td></td>
<td>-0.5608</td>
<td>-0.3047</td>
<td>-0.2246</td>
<td></td>
</tr>
<tr>
<td>230.974</td>
<td>0.0802</td>
<td>0.0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.046</td>
</tr>
<tr>
<td>235.78</td>
<td>-0.0368</td>
<td>0.0386</td>
<td>-0.1807</td>
<td>0.1202</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.1732</td>
</tr>
<tr>
<td>241.318</td>
<td></td>
<td>0.0386</td>
<td>-0.0714</td>
<td>0.1202</td>
<td>0.0644</td>
<td>0.0478</td>
<td>0.2047</td>
<td>0.1067</td>
<td></td>
</tr>
</tbody>
</table>
The results for implementing the model with fit 2 are shown in Tables 2.13 and 2.14, where the accuracy of these fits is dependent on the group of CFD data to which the initial conditions belong, similar to the observation made for fit 1. Since $\alpha_{sc}^+(p_{in})$ was constructed by fitting the scaling factors found for simulation with $p_{b0} = 218.577$ kPa (31.702 psi) and $\alpha_{sc}^-(p_{in})$ from those for simulation with $p_{b0} = 241.318$ kPa (35 psi), simulations with $p_{b0} = 218.577, 226.313, 235.78$ and $241.318$ kPa (31.702, 32.824, 33, 34.197 and 35 psi) were the most accurate. In this case, however, the shock wave propagated further downstream than expected in simulation with the other group of initial conditions because $\alpha_{sc}^+(p_{in})$ overestimated $\alpha_{sc}$; near the lower end of the pressure range of interest, $\alpha_{sc}$ overestimated the fit by over 100%.

Fit 2 improved the reliability of the model when simulated for a decrease in back-pressure, as only 4 simulations terminated due to numerical errors: the two simulations which did not run successfully for any simulation, and simulations where pressure decreased from $p_{b0} = 227.528$ kPa (33 psi) to $p_{in} = 220.632$ kPa (32 psi) and from $p_{b0} = 235.78$ kPa (34.197 psi) to $p_{in} = 226.313$ kPa (32.824 psi). The improved reliability is related to the fact that $\alpha_{sc}^-(p_{in})$ was more likely to underestimate $\alpha_{sc}$ for fit 2 than it was for fit 1.
Table 2.13: Shock location errors relative to the isolator duct height $D_H = 2R(z)$, where $R(z)$ is the radius of the isolator, for simulation of the nonlinear model in equation (2.101) with $\alpha_{sc}^\pm(p_{in})$ from fit 2, with coefficients in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{in}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td>1.9009</td>
<td>3.7417</td>
<td>2.1937</td>
<td>3.2727</td>
<td>3.1253</td>
<td>0.5354</td>
<td>1.164</td>
<td>1.1027</td>
</tr>
<tr>
<td>213.737</td>
<td>1.6218</td>
<td></td>
<td>-3.8744</td>
<td>-2.2564</td>
<td>-3.3309</td>
<td>-3.1253</td>
<td>-0.5354</td>
<td>-1.164</td>
<td>-1.1027</td>
</tr>
<tr>
<td>218.577</td>
<td>-0.2154</td>
<td>-0.6256</td>
<td></td>
<td>0.1342</td>
<td>-1.2844</td>
<td>-1.15</td>
<td>1.2842</td>
<td>0.4572</td>
<td>-0.3381</td>
</tr>
<tr>
<td>220.632</td>
<td>-1.6218</td>
<td>-1.7638</td>
<td>-2.9272</td>
<td></td>
<td>-3.3309</td>
<td>-3.1253</td>
<td>-0.5354</td>
<td>-1.164</td>
<td>-1.1027</td>
</tr>
<tr>
<td>226.313</td>
<td>-1.4903</td>
<td>-1.8323</td>
<td>-2.0321</td>
<td>-0.6764</td>
<td></td>
<td>-1.2176</td>
<td>1.2842</td>
<td>0.4572</td>
<td>-0.3381</td>
</tr>
<tr>
<td>227.528</td>
<td>-0.4405</td>
<td>-1.4907</td>
<td>-1.692</td>
<td></td>
<td>-1.1486</td>
<td></td>
<td>1.8154</td>
<td>0.9222</td>
<td>-0.0575</td>
</tr>
<tr>
<td>230.974</td>
<td>-3.1242</td>
<td>-1.7638</td>
<td>-3.133</td>
<td>-2.6243</td>
<td></td>
<td></td>
<td>-1.164</td>
<td>-1.1027</td>
<td></td>
</tr>
<tr>
<td>235.78</td>
<td>-1.4903</td>
<td>-1.8323</td>
<td>-3.0645</td>
<td>-0.4053</td>
<td></td>
<td>-2.2084</td>
<td>0.5015</td>
<td>-0.3381</td>
<td></td>
</tr>
<tr>
<td>241.318</td>
<td>-0.5565</td>
<td>-1.558</td>
<td>-2.8584</td>
<td>-0.2699</td>
<td>-2.7909</td>
<td>-2.8303</td>
<td>-0.1219</td>
<td>-0.319</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.14: Relative errors in estimated $\alpha_{sc}$, calculated with respect to the expected values in Table 2.7, for simulation of the model with $\alpha_{sc}^+(p_{in})$ defined from the coefficients for fit 2 in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{b0}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>213.737</td>
<td>−0.0341</td>
<td></td>
<td>1.0079</td>
<td>0.7434</td>
<td>0.6471</td>
<td>0.1232</td>
<td>0.2332</td>
<td>0.3824</td>
<td></td>
</tr>
<tr>
<td>218.577</td>
<td>−0.0740</td>
<td>−0.0743</td>
<td>0.2833</td>
<td>0.2463</td>
<td>−0.2919</td>
<td>−0.0663</td>
<td>0.0927</td>
<td></td>
<td></td>
</tr>
<tr>
<td>220.632</td>
<td>−0.0341</td>
<td>0.0062</td>
<td>0.0430</td>
<td></td>
<td></td>
<td>0.1232</td>
<td>0.2216</td>
<td>0.3824</td>
<td></td>
</tr>
<tr>
<td>226.313</td>
<td>−0.0054</td>
<td>0.0449</td>
<td>−0.2467</td>
<td></td>
<td></td>
<td>0.2353</td>
<td>−0.2919</td>
<td>−0.0663</td>
<td>0.0927</td>
</tr>
<tr>
<td>227.528</td>
<td>−0.0054</td>
<td>0.0449</td>
<td>−0.2024</td>
<td>−0.2744</td>
<td></td>
<td>−0.3968</td>
<td>−0.1567</td>
<td>0.0249</td>
<td></td>
</tr>
<tr>
<td>230.974</td>
<td>−0.0054</td>
<td>0.0867</td>
<td>−0.0649</td>
<td>−0.0390</td>
<td></td>
<td></td>
<td></td>
<td>0.3824</td>
<td></td>
</tr>
<tr>
<td>235.78</td>
<td>−0.0054</td>
<td>0.0449</td>
<td>−0.2024</td>
<td>0.0764</td>
<td>0.0540</td>
<td></td>
<td></td>
<td></td>
<td>0.0927</td>
</tr>
<tr>
<td>241.318</td>
<td>−0.0054</td>
<td>0.0449</td>
<td>−0.0961</td>
<td>0.0764</td>
<td>−0.0120</td>
<td>−0.0345</td>
<td>0.0886</td>
<td>−0.0334</td>
<td></td>
</tr>
</tbody>
</table>
The results in Tables 2.15 and 2.16 reflect the fact that fit 3 lies between fits 1 and 2, specifically for the case of an increase in backpressure, as the distinction between the two groups of CFD data is not as clear through these errors. While the results for simulations with $p_{b0} = 218.577$ kPa (31.702 psi) and $p_{b0} = 235.78$ kPa (34.197 psi) were the most accurate of all the fits, the errors were generally in-between those calculated for the other two fits. There were just three cases where little shock propagation was observed since fit 3 did not underestimate $\alpha_{sc}$ as much as fit 1 did; the inclusion of scaling factors from both groups of data also helped reduce the estimation error for the cases where fit 2 overestimated $\alpha_{sc}$ by over 100%. The number of simulations with a decrease in backpressure that terminated increased to 12 for this fit due to the large overestimation of $\alpha_{sc}$ for a small range of inputs. In particular, when the model was simulated with $p_{b0} = 230.974$ kPa (33.5 psi) and when $p_{in} = 218.577$ kPa (31.702 psi), four simulations (out of the six possible simulations) terminated due to numerical errors introduced by the inaccurate fit.
Table 2.15: Shock location errors relative to the isolator duct height $D_H = 2R(z)$, where $R(z)$ is the radius of the isolator, for simulation of the nonlinear model in equation (2.101) with $\alpha_{sc}^\pm(p_{in})$ from fit 3, with coefficients in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{in}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td>0.6</td>
<td>-2.3757</td>
<td>-0.8794</td>
<td>-2.4763</td>
<td>-2.4618</td>
<td>-0.4194</td>
<td>-1.3895</td>
<td>-0.6063</td>
<td></td>
</tr>
<tr>
<td>213.737</td>
<td></td>
<td>-2.4446</td>
<td>-0.8794</td>
<td>-2.4763</td>
<td>-2.4618</td>
<td>-0.4194</td>
<td>-1.3895</td>
<td>-0.6063</td>
<td></td>
</tr>
<tr>
<td>218.577</td>
<td>-0.2154</td>
<td>-0.6256</td>
<td></td>
<td>1.9413</td>
<td>-0.0665</td>
<td>-0.2014</td>
<td>1.4167</td>
<td>0.1295</td>
<td>0.5369</td>
</tr>
<tr>
<td>220.632</td>
<td>-1.6218</td>
<td>-1.6954</td>
<td></td>
<td></td>
<td>-2.54</td>
<td>-2.4618</td>
<td>-0.4194</td>
<td>-1.3895</td>
<td>-0.6063</td>
</tr>
<tr>
<td>226.313</td>
<td>-0.7951</td>
<td>-0.6256</td>
<td>-1.8275</td>
<td></td>
<td></td>
<td>-0.2014</td>
<td>1.4167</td>
<td>0.1295</td>
<td>0.5369</td>
</tr>
<tr>
<td>227.528</td>
<td>-0.4405</td>
<td>-0.0605</td>
<td></td>
<td></td>
<td>0</td>
<td>-1.2165</td>
<td></td>
<td>2.0145</td>
<td>0.5895</td>
</tr>
<tr>
<td>230.974</td>
<td>-3.1242</td>
<td>-1.6954</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-1.3895</td>
<td>-0.6063</td>
</tr>
<tr>
<td>235.78</td>
<td>-1.425</td>
<td></td>
<td></td>
<td>-0.4053</td>
<td>-1.8894</td>
<td>-1.8842</td>
<td>-0.1822</td>
<td></td>
<td>0.5369</td>
</tr>
<tr>
<td>241.318</td>
<td>-0.5565</td>
<td>-2.8584</td>
<td>-0.0673</td>
<td></td>
<td></td>
<td>-1.6199</td>
<td>-0.8172</td>
<td>-0.3818</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.16: Relative errors in estimated $\alpha_{sc}$, calculated with respect to the expected values in Table 2.7, for simulation of the model with $\alpha_{sc}^{\pm}(p_{in})$ defined from the coefficients for fit 3 in Table 2.9.

<table>
<thead>
<tr>
<th>$p_{60}$ (kPa)</th>
<th>$p_{in}$ (kPa)</th>
<th>210.29</th>
<th>213.737</th>
<th>218.577</th>
<th>220.632</th>
<th>226.313</th>
<th>227.528</th>
<th>230.974</th>
<th>235.78</th>
<th>241.318</th>
</tr>
</thead>
<tbody>
<tr>
<td>210.29</td>
<td></td>
<td>-0.0034</td>
<td>0.2284</td>
<td>0.1969</td>
<td>0.3601</td>
<td>0.3642</td>
<td>0.0857</td>
<td>0.303</td>
<td>0.1359</td>
<td></td>
</tr>
<tr>
<td>213.737</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>218.577</td>
<td>-0.0749</td>
<td>-0.4482</td>
<td>0.0012</td>
<td>0.0322</td>
<td>-0.3155</td>
<td>-0.0134</td>
<td>-0.1022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>220.632</td>
<td>-0.035</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>226.313</td>
<td>-0.0064</td>
<td>0.08</td>
<td>-0.1973</td>
<td></td>
<td></td>
<td>0.0231</td>
<td>-0.3155</td>
<td>-0.0134</td>
<td>-0.1022</td>
<td></td>
</tr>
<tr>
<td>227.528</td>
<td>-0.0064</td>
<td>0.08</td>
<td>-0.2211</td>
<td></td>
<td></td>
<td>-0.4169</td>
<td>-0.1089</td>
<td>-0.1579</td>
<td></td>
<td></td>
</tr>
<tr>
<td>230.974</td>
<td>-0.0064</td>
<td>0.1232</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1359</td>
</tr>
<tr>
<td>235.78</td>
<td>-0.0064</td>
<td></td>
<td>0.1553</td>
<td>0.0213</td>
<td>0.1265</td>
<td></td>
<td></td>
<td></td>
<td>-0.1022</td>
<td></td>
</tr>
<tr>
<td>241.318</td>
<td>-0.0064</td>
<td>-0.0367</td>
<td>0.1553</td>
<td>0.0319</td>
<td>0.1351</td>
<td>-0.0603</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Since the simulation results do not readily suggest the most accurate fits for $\alpha_{sc}^\pm(p_{in})$, it is necessary to consider trade-offs between accuracy and reliability of the model. The latter consideration is only necessary in evaluating the fits for $\alpha_{sc}^-(p_{in})$, as these are the only simulations where numerical errors are encountered. Although fit 3 produces the most accurate results, in terms of relative error in steady-state shock location, this fit also leads to the largest number of failed simulations. If it were possible to improve implementation to avoid these errors, fit 3 would be the best choice, but given the data presented here, fit 2 would be the best choice for $\alpha_{sc}^-(p_{in})$ as it presents the fit for which simulations were least likely to terminate due to numerical errors.

An argument can be made that fit 3 is the best fit for $\alpha_{sc}^+(p_{in})$, even though there are specific simulations where fits 1 and 2 are more accurate, because it represents a ‘middle ground.’ In this case, fit 3 is not necessarily the ‘most accurate’ fit but the fit that produces the fewest number of cases where the shock location error is large and inconsistent with the two other fits.

The characterization of the isolator model in equation (2.101) with the choice of fit 3 for $\alpha_{sc}^+(p_{in})$ and fit 2 for $\alpha_{sc}^-(p_{in})$ (with coefficients in Table 2.9) is the most accurate physics-based model available. Inaccuracies remain in the model, which may be improved by constructing scaling functions that better estimate $\alpha_{sc}$. This may be done by collecting a new set of steady-state CFD solutions representing cases where the shock is more regularly spaced in the isolator. The results discussed here suggest that $\alpha_{sc}$ is not dependent on the initial condition, so separate functions for $\alpha_{sc}^+(p_{in})$ and $\alpha_{sc}^-(p_{in})$ may be constructed by considering the scaling factors in sets that depend only on the change in backpressure.

In order to be useful in a control design application, the model must also be shown to be temporally accurate; that is, to correctly capture the transient behavior of the
shock wave. This is not the case of the PBM as it takes 20 ms longer for a steady-state condition to be established. After the spatial accuracy of the model has been verified through simulation with single-step inputs, the model should be examined to discover the source of the temporal inaccuracy and validated against cases with a multi-step or continuous input.

The model developed by Chicatelli and Hartley in Refs. 28, 29 has been improved through the damping and scaling factor modifications introduced in this chapter, but its usefulness for control design must be carefully vetted until the accuracy of the scaling functions can be improved and the temporal accuracy addressed.
Chapter 3: Data-based model

Concurrent with the nonlinear physics-based modeling effort described in Chapter 2, a linear data-based model (DBM) of the isolator was built using a ‘black box’ subspace identification method [43–45], which utilizes CFD simulation data in constructing the model. To the best of the author’s knowledge, this methodology has not yet been applied to scramjet modeling efforts. Here, a small-order, single-input, single-output (SISO) linear model of the isolator is constructed using the subspace state-space (4SID) model identification technique developed by van Overschee and de Moore [44]. A set of models was constructed and validated to determine the accuracy of each model; for large-order models, a set of reduced-order models was also constructed. Through validation, a set of candidate models was chosen for use in developing a model-based controller.

Because the closed-loop model, with a controller designed to anchor the shock wave in the presence of input (flow) perturbations, will be analyzed for robust stability, it is necessary to be able to show the plant is robust against parameter perturbations. Uncertainty in the isolator model, introduced by modeling errors, may have either a structured or unstructured form [46,47]. Analysis of model robustness is facilitated by representing uncertainty in a multiplicative form, so such a representation is developed for each candidate model. This analysis suggests a single candidate model of the isolator can be selected, for which control design and analysis was completed.
3.1 Modeling methodology

A control-oriented model (COM) of the isolator is necessary to balance the accuracy and size of the model so that the model-based controller designed to prevent unstart during periods of high engine demand is of a small order. When the isolator model is cascaded with a combustor model, the closed-loop system has the structure in Figure 3.1, where details of the DBM have been added to Figure 1.3 to reflect the linear SISO structure and identify the input (\(u\)) and output (\(y\)) of the model. The input and output are offset by steady-state values of the backpressure, \(p_{b,\text{offset}}\), and shock location, \(z_{sh,\text{offset}}\), to improve accuracy of the model during identification.

In this effort, the N4SID model identification technique is implemented, using transient CFD data, to construct the isolator model \([43, 44] \). As shown in Figure 3.1, this identification process yields an LTI state-space model with the form

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\] (3.1)

where \(x \in \mathbb{R}^n\), \(u \in \mathbb{R}^m\), \(y \in \mathbb{R}^q\), and the matrices \(A\), \(B\), \(C\), and \(D\) have compatible dimensions. For this SISO model, \(m = q = 1\) and the system input, \(u\), is the back-pressure imposed on the isolator exit plane and the system output, \(y\), is the location.
of the shock wave in the isolator. Both the input and output of the model are measured as perturbations from the steady-state offsets $p_{b,\text{offset}}$ and $z_{sh,\text{offset}}$, respectively, as these offsets are subtracted from the CFD data before identification in order to remove bias from the resulting model.

The model identification process has several steps, as detailed by Ljung [43]: design of an experiment, collection of data, model identification and model validation.

### 3.1.1 Data Collection and Preprocessing

The design of a suitable set of experiments for data collection is one of the most important steps of system identification. This involves designing an input that excites a sufficiently large number of modes in the system, which allows for a model to be constructed from the input signal and output measurements that accurately estimates the input-output dynamics of the system [43].

In addition to the guidelines presented in Ref. 43, the set of acceptable inputs is limited by the CFD solver. The most stringent constraint was due to the fact that the CFD simulations were run for inviscid flow through the isolator, where the lack of a viscous mechanism necessitated that the input signal was a non-decreasing function in order to limit the non-physical phenomena observed during simulation. CFD simulations were run for inviscid flow to be able to compare results for the DBM (constructed through system identification) to the PBM (constructed from the Euler equations). Because of this limitation, the DBMs could not be validated for decreases of backpressure, since those dynamics were not present in the identification data and, therefore, were not captured in the models. These dynamics could be added later if identification data sets were collected from additional CFD simulations. Three sets of data were collected:
1. *stepA*: shock dynamics due to an increase of backpressure, with the shock wave settling to a steady-state location at the end of the experiment (identification data)

2. *stepB*: shock motion due to an input signal with the same initial and final values, but different transitional behavior, as the identification data set (validation data)

3. *stepC*: shock motion due to an input with a final value perturbed from that of the identification data set (validation data)

The inputs for each of these experiments are shown in the top row of Figure 3.2, where the initial value of backpressure ($p_0 = 213.737$ kPa = 31 psi) allows for a shock wave to form and establish a non-steady-state location near the isolator exit. The final value of pressure in experiments ‘stepA’ and ‘stepB’ ($p_f = 227.528$ kPa = 33 psi) establishes a shock wave near the middle of the isolator in steady-state, as can be seen from the profiles in Figure 3.3. In experiment ‘stepC,’ the final value of pressure ($p_f = 241.318$ kPa = 35 psi) establishes a steady-state shock location further upstream, as Figure 3.4 shows. It can be seen from Figure 3.2 that the input for experiment ‘stepA’ is a uniform-step function, with step-height 6.895 kPa (1 psi) and step duration 15 ms, while the inputs for the other two experiments were designed as variable-step signals expected to excite different modes than the uniform-step function.

Propagation of the shock wave in each experiment is shown in the bottom row of Figure 3.2. The shock position was calculated from the pressure profiles by determining the location along the isolator where pressure rose above 10% of the tare value ($p/p_{\text{tare}} = 1.1$). The accuracy of this algorithm was verified graphically by comparing the estimated locations to the profiles from CFD (the dashed and solid lines in Figures 3.3 and 3.4, respectively).
Figure 3.2: Input and output signals, sampled at $T_s = 0.5$ ms in CFD simulation, used for system identification and validation (prior to detrending).

Figure 3.3: Pressure profiles at the isolator wall for experiment ‘stepA’ (uniform-step input) with estimated shock wave locations indicated by dashed lines and the isolator boundaries by thick black lines. Profiles for experiment ‘stepB’ (variable-step input) show establishment of a shock wave at the same steady-state location, but in a different transient manner.
Figure 3.4: Pressure profiles at the isolator wall for experiment ‘stepC’ (variable-step input) with estimated shock wave locations indicated by dashed lines and the isolator boundaries by thick black lines.

Ljung suggests that the data sets be ‘detrended’ before proceeding with model identification in order to remove bias from the model [43]. This is done through removal of a mean or a given off-set, such as a steady-state condition, from the input and output signals. The experiments run for this modeling effort were designed such that the shock wave established a steady-state location at the end of the experiment, and so the final input and output values of the identification data could be used as the off-sets. Here, two families of models are constructed: one using the input-output data from experiment ‘stepA’ as identification data and one using the data from experiment ‘stepB.’ The off-sets in Table 3.1 are removed from each set of data before construction and validation, so that $u$ and $y$ in equation (3.1) are perturbations from these steady-state values.
Table 3.1: Input and output off-sets used for detrending the CFD simulation data for identification of models in each family.

<table>
<thead>
<tr>
<th>Identification Data</th>
<th>Input Offset</th>
<th>Output Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepA</td>
<td>227.528 kPa (33 psi)</td>
<td>0.6407 m</td>
</tr>
<tr>
<td>stepB</td>
<td>227.527 kPa (33 psi)</td>
<td>0.64838 m</td>
</tr>
</tbody>
</table>

3.1.2 Model Identification and Reduction

After data collection and preprocessing, the model identification algorithm can be implemented. In this effort, LTI models were constructed using the N4SID algorithm in the MATLAB® System Identification Toolbox (SIDT). The uniform-step input was designed to generate ‘identification data’ for the model and the variable-step data sets were designed to be used for model validation, but because the inputs for experiments ‘stepA’ and ‘stepB’ share initial and final values, a second ‘family’ of models was constructed by reversing the roles of the data sets. This offered a third variable, in addition to the model order \( n \) and input delay (relative degree) \( nk \), to consider in constructing the models. In the following discussions, the two model families are distinguished by the set of data used for identifications: ‘stepA’ for models constructed from the data produced from the uniform-step input and ‘stepB’ for those constructed from the data produced from the variable-step input.

The structure of the state-space model is affected by the choice of input delay \( nk \). When a model does not have an input delay, \( nk = 0 \), the feedthrough matrix \( D \) in equation (3.1) is nonzero and the output has direct dependence on the input. However, when \( nk > 0 \), \( D = 0 \) and the output function must be differentiated \( nk \) times before the input appears. When \( nk > 1 \), a state vector with dimension \( (nk - 1) \),
is augmented to the system, resulting in the model

\[
\begin{bmatrix}
\dot{x} \\
\dot{x}_a
\end{bmatrix} =
\begin{bmatrix}
A_{11} & A_{12} \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
x_a
\end{bmatrix} +
\begin{bmatrix}
0 \\
B_a
\end{bmatrix} u
\]

(3.2)

\[y = [C_1 \ 0] \begin{bmatrix} x \\ x_a \end{bmatrix}\]

The models of interest have delay order \( nk \leq 2 \), and the structure of the system for \( nk = 0 \) and \( nk = 1 \) is given in equation (3.1), while equation (3.2), with \( B_a = 1 \) and \( \dot{x}_a = u \), gives the structure when \( nk = 2 \).

**Model Identification**

The SIDT has available a graphical user interface (GUI) that allows for quick construction and validation of a model to help determine suitable choices for model order and delay. Seventy-three preliminary models were constructed using this tool: 45 in the ‘stepA’ family and 28 in the ‘stepB’ family. An \((n, nk)\) pair was associated with each model, indicating the model order and input delay. The delay variable \( nk \) took on values in the discrete range \([0, 10]\), where the upper bound was the value at which the accuracy of each model noticeably decreased.

For each \( nk \), the maximum size of the model is dependent on the number of samples in the identification data. The maximum order of the ‘stepA’ family of models is higher \((n = 45)\) than that of the ‘stepB’ family \((n = 32)\) since the uniform-step input is longer than the variable-step input; for both model families, the smallest model constructed had order \( n = 5 \), with model orders increasing by 5 to the maximum value for a given delay \( nk \) and identification data set.
Figure 3.5: Dependence of the ‘best fit’ parameter (equation (3.3)) on the model order, labeled by delay \( nk \) and identification data (A or B), for simulation with inputs from experiments ‘stepA,’ ‘stepB,’ and ‘stepC.’

The GUI provides quantification of the model fit through a calculation that depends on results from simulation of each model with each input:

\[
\text{B.F.} = (1 - \frac{|\hat{y} - y|}{|y - \bar{y}|}) \cdot 100 \tag{3.3}
\]

where \( y \) is the output predicted from the CFD data, which has mean \( \bar{y} \), and \( \hat{y} \) is the output estimated from simulation of the model. Calculations of this parameter for models validated against each multi-step input are shown in Figure 3.5, which can be used to help determine a set of models for closer consideration. The fit calculations for ‘stepA’ (‘stepB’) models validated against experiment ‘stepA’ (‘stepB’) data are included only for completeness, as it is expected for a model to show high accuracy when validated against the data used to identify the model.
It can be seen from Figure 3.5 that models with larger input delays \((nk = 5\) and \(nk = 10\)) are markedly less accurate than those with smaller delays, most notably for the ‘stepB’ family of models. These results reveal that larger-order models are not necessarily the most accurate because of the potential for over-fitting a model to the identification data. For example, when \(nk = 1\), a model with order \(n = 5\) is at least as accurate as a larger-order model \((nk = 45)\). When \(nk = 0\), however, model accuracy seems to trend upward as model order increases as the additional states capture more dynamics in the model.

In addition to these empirical calculations, the Akaike information criterion (AIC), which estimates the amount of ‘information’ captured by a model and is related to the likelihood function of the model, can also guide the choice of model order. The AIC includes a penalty based on the size of the parameter set to avoid over-fitting the data and improve the accuracy of the model when simulated with other inputs. Calculation of AIC in MATLAB\textsuperscript{®} uses the loss function \(V\), which depends on the prediction error \(\epsilon(t, \theta_N)\) and the negative log of the likelihood function, and enforces an over-fitting penalty dependent on the size of the identification data set, \(N\), and the number of parameters to be estimated, \(d\):

\[
AIC = \log V + \frac{2d}{N} = \log \left| \frac{1}{N} \sum_{t=1}^{N} \epsilon(t, \theta_N)\epsilon^T(t, \theta_N) \right| + \frac{2d}{N} \tag{3.4}
\]

For each family of models, the AIC is calculated for models with input delays \(nk \in [0\ 10]\) and orders ranging from \(n = 1\) to \(n_{\text{max}}\). These results are shown in Figure 3.6.

The effect of large input delay in the system (large \(nk\)) can be observed in Figure 3.6, as the AIC increases noticeably when \(nk > 3\). The AIC calculations for the ‘stepA’ model family are ‘smoother’ than those for the ‘stepB’ family, with values
reaching a minimum around $n = 5$ (for $nk = 0$ and $nk = 1$) before increasing as $n$ increases. For the ‘stepB’ model family, the AIC does not have as clear a trend as the calculations show a minimum around $n = 7$, but also exhibit large spikes for models with higher orders. It is also interesting to note that models in the ‘stepA’ family with $nk = 0$ and $nk = 1$ have better AIC for nearly all model orders than models with larger input delay, but those same models in the ‘stepB’ family have better AIC only up to order $n = 22$, when the models with input delays $nk = 2$ and $nk = 3$ have better AIC, suggesting that it is necessary for higher-order models to have larger relative degree in order to better capture the input-output dynamics of the system.

Limiting the size of the models by considering only those with $nk < 3$, it can be seen that the AIC are generally lower for the ‘stepA’ family of models than for models in the ‘stepB’ family, suggesting that models constructed using ‘stepA’ data are expected to be more accurate. Because the range of AIC for both model families
is within 2 of the minimum for model orders $n > 1$, it can also be expected that models within the same family include enough information to accurately model the system dynamics.

Because the goal of this effort is to construct a COM, it is not necessarily a disadvantage for a model to have a small order if it is sufficiently accurate; this can be achieved at the level of system identification or through model reduction. Both approaches were considered for models with $nk = 0$ and $nk = 1$, and the accuracy of a small-order DBM was compared to that of a small-order model obtained through reduction of a larger-order DBM.

Taking into consideration the best-fit and AIC calculations, five full-scale models (FSMs) were constructed for each model family. In some cases, the model size was chosen to balance conclusions drawn from the two calculations because the prediction errors from which AIC is calculated depend on the identification data, suggesting that additional analysis, such as the empirical ‘best fit’ data, should supplement the AIC analysis. Trade-offs are necessary to consider when AIC calculations suggest a smaller-order model to be more accurate, while best-fit analysis may suggest a larger model to be more accurate. For example, models built with input delay $nk = 2$ had orders chosen with more emphasis placed on the empirical fit calculations, despite AIC results. This motivated the choice of $n = 35$ for the ‘stepA’ family model and $n = 25$ for the ‘stepB’ family model; small-order models with delay $nk = 2$ were not considered because the additional state augmented when $nk = 2$ may be an unnecessary complication when working with small-order models and this analysis suggests that smaller-order models with $nk < 2$ are more accurate.

Two models each with input delay $nk = 0$ and $nk = 1$ were constructed for both the ‘stepA’ and ‘stepB’ families: a small order model that might be used for control design and a large order model that should be reduced before proceeding with design.
Table 3.2: Model delays \((nk)\) and orders \((n)\) selected for construction of data-based models using CFD data, given for each model family considered.

<table>
<thead>
<tr>
<th>(nk)</th>
<th>(n)</th>
<th>(nk)</th>
<th>(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepA</td>
<td>0</td>
<td>15, 30</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5, 30</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.2 lists the orders of the FSMs constructed. Note that for the \(nk = 1\) case, the order \(n = 5\) model had the lowest AIC of all models considered and is expected to be among the most accurate models.

Each model in Table 3.2 was constructed using the N4SID algorithm built-in to MATLAB®, where properties of the model, such as how the initial conditions are treated and whether a noise model should be estimated, were specified when implementing the function. The covariance matrix calculated during identification provides an estimate of the standard deviation of the matrix parameters (when in controllability canonical form) that can be used to define model of the isolator containing a representation of uncertainty.

**Model Reduction**

The size of a model-based controller, and the complexity of the closed-loop system, is limited by defining a small-order COM for use in design. Some of the FSMs in Table 3.2 are small enough to be immediately applicable as COMs, while others require reduction to find a COM of appropriate order. One of several model order reduction techniques can be implemented to reduce the size of the larger FSMs [48].

Balanced-truncation is a well-established technique that involves expressing the model in a coordinate system that allows for identification of the most-controllable and most-observable states by the Hankel singular values (HSV) of the system (that is,
Figure 3.7: Example Hankel singular value plots, for models in the ‘stepA’ family with $nk = 1$. The singular values indicate which states contribute most to the input-output dynamics: less-significant states can be truncated with limited loss of accuracy.

the singular values of the product of the controllability and observability grammians). The singular value decomposition (SVD) of the product of the grammians can be used to measure the relative contribution of each state in a balanced realization to the input-output dynamics of the system. States with lower contribution (smaller HSV) are less significant in the system realization and can be truncated, introducing an error with a known upper bound [48]. Example HSV plots for three models (selected from those in Table 3.2) are shown in Figures 3.7 and 3.8.

Figure 3.7 contains plots of the HSVs for the models in the ‘stepA’ family with input delay $nk = 1$. Although a single state appears to capture most of the system energy for the model with order $n = 5$ and delay $nk = 1$, previous work has shown that such a model would likely not capture relevant dynamics, suggesting the reduced-order model (ROM) should have an order $k > 1$. For this FSM, two ROMs are constructed: one with order $k = 3$ and one with order $k = 4$, the model orders where the singular
Table 3.3: Orders of FSMs and the associated ROMs constructed to model the scram-
jet isolator.

<table>
<thead>
<tr>
<th>$nk$</th>
<th>$n$</th>
<th>$k$</th>
<th>$nk$</th>
<th>$n$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepA</td>
<td>0</td>
<td>15</td>
<td>3, 4</td>
<td>stepB</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>30</td>
<td>2, 4, 5, 6</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>3, 4</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>30</td>
<td>2, 4, 5, 6</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>35</td>
<td>2, 5</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

value magnitude decreases. Although these reductions are unnecessary for this FSM, analysis of these ROMs is included for completeness.

The idea of guiding the choice of reduced model order $k$ using the plots of the HSVs can be extended to the larger-order FSMs, where more ROMs may be investigated since the HSVs show more relative changes in magnitude, as can be seen in Figure 3.7. Two ROMs are constructed for the smaller-order models with delays $nk = 0$ and $nk = 1$ in the ‘stepA’ family, while four ROMs are considered for each of the large-order FSMs. The decrease in magnitude between HSVs for the ‘stepB’ family of models is much larger, suggesting that fewer states are necessary to capture the dynamics, so only two ROMs are constructed for each FSM.

An extra state is added to the system when $nk = 2$, as shown in equation (3.2); the HSVs for the FSM in the ‘stepB’ family with $nk = 2$ are plotted in Figure 3.8. The HSVs exhibit two more-defined ‘drops’ in magnitude for the ‘stepA’ family of models, motivating the choice of two reduced orders, $k = 2$ and $k = 5$. The HSV plot of the ‘stepB’ family model suggest a set of four reduced orders $k$ to consider. Table 3.3 lists the orders of the ROMs constructed for each FSM.

Balanced truncation was implemented using built-in MATLAB® routines to construct the set of ROMs with the orders $k$ in Table 3.3. Each FSM was balanced, the most-controllable, most-observable states identified, and the states associated with
small HIVs truncated to form each ROM. This yields a set of 26 ROMs: 14 constructed from ‘stepA’ family FSMs and 12 from ‘stepB’ family FSMs. A collection of 36 models was built, with 19 models in the family constructed from the identification data with uniform-step input and 17 in the family constructed using the identification data with variable-step input.

The stability of each FSM and ROM was verified before validation. Each FSM was constructed to be stable, and the ROMs were expected to be stable, since balanced truncation preserves stability [48], however it is useful to check this in case errors may have occurred. The eigenvalue spectra for a candidate model set are shown in Figure 3.9, where the set includes one ROM (the order $k = 5$ model reduced from the order $n = 30$ model in the ‘stepA’ family with input delay $nk = 1$); the unit circle represents the stability region for discrete-time models. As expected, each of the 36 models was found to be stable.
Figure 3.9: Eigenvalues of candidate models of the isolator. Each model shown here is stable, and it was found that all models considered (Table 3.3) were stable.

### 3.2 Model validation

Validation of each DBM allows for verification that the model captures the dynamics of the system due to the input used to generate the identification data and due to input perturbations. If the model does not accurately capture the dynamics due to input perturbations, it will not be useful in control design. The three data sets described in Section 3.1.1 are used for validation of each model constructed in this effort, where validation against identification data is performed to ensure accurate reproduction of this data.

The input signals of the two experiments used to construct the identification data sets, ‘stepA’ and ‘stepB,’ had the same initial value and final value, and the steady-state shock wave locations for the two experiments were within 8 mm of each other (as shown in Table 3.1). The final value of the input was perturbed in the third
experiment (‘stepC’), resulting in a steady-state shock location significantly different from the other experiments.

Each model was simulated using functions in the SIDT, where initial conditions were estimated from the validation data to improve the overall accuracy of the models. This choice was suggested from the results of previous work with DBMs, although further investigation to study the effect of the choice of initial condition on model accuracy may be done, if necessary.

3.2.1 Results

Because each of the 36 models in Table 3.3 was simulated for three different inputs, it is impossible to present the full set of simulation results here; instead, results are presented for a set of candidate models chosen to be representative of the complete model set. The candidate set contains four models: two FSMs from the ‘stepA’ family, with input delay $nk = 1$, one of a small order ($n = 5$) and one of a large order ($n = 30$); the order $k = 5$ ROM obtained through reduction of the order $n = 30$ model; and the FSM from the ‘stepB’ family with input delay $nk = 2$ and order $n = 25$. These are the most accurate models in each family, as determined through evaluation of the empirical fit parameter and AIC along with output and simulation analysis.

Graphical results for simulation of this model set are presented in Figures 3.10, 3.12 and 3.14, where the simulation output and predicted output (calculated from the CFD data) are shown together in the top plot of each figure, and the output error is shown in the bottom plot. Three error calculations are made for each simulation (root-mean-square (RMS), minimum, and maximum errors) and presented in Tables 3.4 to 3.6, offering numerical analysis of the results. Note that the outputs plotted in the figures
have not been retrended using the information in Table 3.1, and therefore represent
the offsets of the shock wave from the steady-state location in the identification data.

For each simulation, residual analysis was also performed. Auto-correlation of the
output errors and cross-correlation between the input and the output errors are shown
in Figures 3.11, 3.13, and 3.15, where the 99% confidence intervals (CI) are indicated
by boxes. Auto-correlation of the output errors acts as a ‘whiteness test’ for the model
as the calculations help to determine if the present output has dependence on past
outputs. It is expected, and seen in each plot, that the (normalized) auto-correlation
is 1 at a time delay of 0 for each model, and will ideally be within the CI for delays
greater than 0. Cross-correlation between the input and output provides evaluation
of the choice of input delay \( n_k \); if a peak appears with a magnitude outside the CI,
it may indicate that the output has dependence on a past input, which suggests that
a larger input delay be considered. For these models, the cross-correlation remained
inside the CI for all simulations, suggesting the respective choice of delay \( n_k \) to be
adequate.

The first set of results presented here consider validation of the models against the
experimental data obtained through simulation with the uniform-step input (the first
plot in Figure 3.2); this represents validation against the identification data for mod-
els in the ‘stepA’ family, where these models were found to accurately reconstructed
the identification data. It is not sufficient, or correct, to use only these simulations
to evaluate model accuracy, so simulation with the two variable-step inputs provides
additional means for evaluation. A similar statement may be made for models in the
‘stepB’ family, where the model is validated against the identification data through
simulation with the variable-step input in the second plot of Figure 3.2. This is impor-
tant to keep in mind when studying the results presented here, where differentiation
is not made between the type of data set against which each model is validated.
Simulation with uniform-step input (experiment ‘stepA’)

Simulation results from validation of the candidate model set against the experimental data generated from simulation with the uniform-step input are presented in Figure 3.10 and Table 3.4. The table gives calculations of the minimum, maximum, and RMS errors, while the figure shows simulation and predicted outputs (top) and the output error (bottom). The order and input delay for each model is indicated in the legend, where it should be noted that the order \( n = 25 \) model is from the ‘stepB’ family and the others are from the ‘stepA’ family.

The high level of accuracy observed through these simulations of the ‘stepA’ model family is expected since the identification and validation data sets are identical. The results from simulation of these models, in Figure 3.10, show that the output errors for both FSMs were under 5 mm, suggesting the extra states in the \( n = 30 \) model do not significantly improve model accuracy. The loss of accuracy expected from model reduction can be seen from simulation of the ROM, where the output error exceeds 2 cm early in the simulation, and remains above 1 cm before establishing a steady-state at the end of the simulation. The peak errors are less than 17.4% of the average duct height, \( \bar{D}_H = 11.5 \) cm; the shock wave is expected to oscillate up to one duct height when in steady-state, so these errors are well within this limit. Although not shown here, the accuracy of models with the same orders, but input delay \( nk = 0 \), are similar, suggesting that this choice of models from the ‘stepA’ family is representative of the entire family of models when \( nk < 2 \).

It should be noted that the smallest ROMs constructed from the order \( n = 30 \) models, which have reduced orders \( k = 2 \) and \( k = 4 \), do not contain enough states to capture the input-output dynamics of the isolator system. A similar observation is made for the \( nk = 2 \) model set in the ‘stepA’ family, where neither ROM accurately models the system. Larger ROMs of the \( nk = 2 \) FSM were not considered because
models with $nk = 0$ and $nk = 1$ were found to be more accurate with fewer states, suggesting that an input delay $nk = 2$ may not be ideal for this system.

In contrast, the ‘stepB’ family of models can be validated through simulation with the uniform-step input since the modes excited by this input differ from those excited by the identification input. Model reduction errors were again observed during analysis of these results, but are not shown in Figure 3.10. For example, when $nk = 0$, the output errors for both the order $n = 10$ and $n = 20$ FSMs were under 9 mm, but the output from simulation of the ROMs had errors ranging between 1 and 2 cm. The order $n = 25$ model with delay $nk = 2$ is more accurate than the other FSMs, with a steady-state error around 8 mm as shown in Figure 3.10, and ROMs constructed from this model were the most accurate in steady-state.

Residual analysis of these simulations is shown in Figure 3.11, where the top plot shows the auto-correlation of the output error, and the bottom plot shows the
Table 3.4: Minimum, maximum and RMS errors for simulation of the candidate model set validated against ‘stepA’ data.

<table>
<thead>
<tr>
<th>nk</th>
<th>n</th>
<th>k</th>
<th>min (m)</th>
<th>max (m)</th>
<th>RMS (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td></td>
<td>9.1749 · 10^{-6}</td>
<td>0.0049581</td>
<td>0.0020108</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td></td>
<td>3.0936 · 10^{-5}</td>
<td>0.004648</td>
<td>0.0018538</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>5</td>
<td>1.0999 · 10^{-5}</td>
<td>0.027567</td>
<td>0.010498</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td></td>
<td>1.4433 · 10^{-15}</td>
<td>0.0077235</td>
<td>0.0026868</td>
</tr>
</tbody>
</table>

cross-correlation of the input with the output error. In both plots, the shaded boxed indicate 99% CIs.

For the ‘stepA’ family of models, the auto-correlation of each FSM was contained inside the 99% CI, except at small time delays for all models. Smaller-order models appear to be ‘whiter,’ as the auto-correlation enters the CI at smaller delays than for larger-order models.

A small amount of correlation is suggested between prediction errors from simulation of the ‘stepB’ family of models with the uniform-step input, indicating that these models are not as white as the ‘stepA’ family. The model with input delay $nk = 2$ passes the whiteness test the best, but does not have comparable results to those of the ‘stepA’ family of models shown in Figure 3.13, which does not leave the CI after entering it.

As previously mentioned, cross-correlation calculations for each FSM and ROM are contained in the CI, and the lack of peaks in these plots (for the FSMs) suggests the choices of delay $nk$ are adequate for these models. Although the cross-correlation calculations for the ROM have more pronounced peaks, which decrease in magnitude as the model order increases, the size of the CI increases so that these calculations remain inside the CI.
Figure 3.11: Residual analysis for validation of the candidate model set against ‘stepA’ data. 99% confidence intervals (CI) are indicated by shaded regions; the auto-correlation calculations had the same CI for each model, but the cross-correlation calculations did not (indicated by boxes shaded with similar color to the line).

Simulation with variable-step input (experiment ‘stepB’)

Results for simulation with the variable-step input from experiment ‘stepB’ are shown in Figure 3.12 and error calculations are presented in Table 3.5. Since this simulation represents validation of the ‘stepB’ model family against the identification data, the results from simulation of the two model families can not be discussed together.

High accuracy is expected from validation of the ‘stepB’ family of models against the identification data, however the output errors increased at time instances when the input experiences a step change. Simulation results for the order $n = 25$ FSM with input delay $nk = 2$, shown in Figure 3.12, have an RMS output error under 6 mm, but the output error exceeds 1 cm ($0.087\bar{D}_H$) at input transitions, but still within one duct height. Unlike the ‘stepA’ family of models, small-order ROMs constructed
Table 3.5: Minimum, maximum, and RMS errors for simulation of the candidate model set validated against ‘stepB’ data.

<table>
<thead>
<tr>
<th>nk</th>
<th>n</th>
<th>k</th>
<th>min (m)</th>
<th>max (m)</th>
<th>RMS (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td></td>
<td>4.5138 \cdot 10^{-7}</td>
<td>0.0060594</td>
<td>0.0024252</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td></td>
<td>2.653 \cdot 10^{-9}</td>
<td>0.0073379</td>
<td>0.0026276</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>5</td>
<td>3.2277 \cdot 10^{-6}</td>
<td>0.026238</td>
<td>0.0096691</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td></td>
<td>1.6653 \cdot 10^{-15}</td>
<td>0.007938</td>
<td>0.0027956</td>
</tr>
</tbody>
</table>

From this model, specifically the ROM with order $k = 2$, were found to accurately capture the input-output dynamics in the system. In fact, this ROM was the most accurate ROM constructed from this FSM.

The output errors for the FSMs with input delay $nk = 0$ and $nk = 1$ were found to be under 5 mm throughout the simulation, but increased to 1 cm in steady-state. The ROMs constructed from these models were much less accurate, with output errors generally between 1 and 2 cm. Although not shown here, it was found that a ROM with order $k = 2$, constructed from the $n = 30$ FSM, was not sufficiently accurate as no system dynamics (shock propagation) were captured, but by adding just one more state ($k = 3$), the error of the ROM was under 5 mm.

Models in the ‘stepA’ family were validated through simulation with the variable-step input from experiment ‘stepB,’ since this differed from the data used to construct these models. As expected, output errors from these simulations were larger than from validation against the identification data, due to the different modes excited by the two inputs. However, the results also suggest that enough similar modes were excited that the accuracy loss was not significant. Results from simulation of the model sets with $nk = 0$ and $nk = 1$ were again similar, so the results in Figure 3.12 represent the set of models in this family with input delay $nk < 2$, where the output errors were under 1 cm for the FSMs, and around 2 cm for the ROMs before becoming more accurate in steady-state.
Correlation calculations for these simulations are shown in Figure 3.13; in general, the auto-correlations are similar to those in Figure 3.11, where the models in the ‘stepA’ family behave better than those in the ‘stepB’ family. The FSMs again pass the whiteness test better than the ROMs.

While the cross-correlation calculations for the FSMs remain inside the CI, pronounced peaks are observed around a time delay of 30 for the ‘stepA’ family and around 100 for the ‘stepB’ family, suggesting that a larger delay might be appropriate for these models. A model with an input delay of \( nk = 30 \) was constructed, but the peak was found at the same location during analysis of this model, suggesting that the present choices of input delay were acceptable. The behavior of the cross-correlations of the ROMs is similar to those in Figure 3.11, and much worse than that of the FSMs.
Figure 3.13: Residual analysis for validation of the candidate model set against the ‘stepB’ data. 99% confidence intervals (CI) are indicated by the shaded regions; the auto-correlation calculations had the same CI for each model, but the cross-correlations did not (indicated by boxes shaded with similar color to the line).

Simulation with variable-step input with perturbed final value (experiment ‘stepC’)

Once validation of the models against a data set with the same steady-state shock location as the identification data was completed, it remained necessary to validate the models against a data set that contained a steady-state shock in a significantly different location. A variable-step input with a final value perturbed 6.895 kPa (1 psi) from that of the uniform-step input was constructed, producing a steady-state shock wave about 35 cm downstream of that in the identification data (experiment ‘stepC’). The simulation results are shown graphically in Figure 3.14 and error calculations are given in Table 3.6.

As expected, the output error is generally larger than for the other validation simulations, most notably at locations of sharp transitions in the expected (CFD)
Table 3.6: Minimum, maximum, and RMS errors for simulation of the candidate model set validated against ‘stepC’ data.

<table>
<thead>
<tr>
<th>nk</th>
<th>n</th>
<th>k</th>
<th>min (m)</th>
<th>max (m)</th>
<th>RMS (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>-</td>
<td>1.7516 \cdot 10^{-8}</td>
<td>0.018962</td>
<td>0.0057484</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>-</td>
<td>2.7294 \cdot 10^{-5}</td>
<td>0.017092</td>
<td>0.0056171</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>5</td>
<td>0.00014001</td>
<td>0.02707</td>
<td>0.011813</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>-</td>
<td>1.2768 \cdot 10^{-15}</td>
<td>0.014463</td>
<td>0.0056865</td>
</tr>
</tbody>
</table>

output, which occur when the input signal changes value. For the model sets with input delays $nk = 0$ and $nk = 1$ in the ‘stepA’ family, which have been shown to behave similarly in simulation, output errors for the FSMs reach nearly 2 cm at these locations, while remaining under 1 cm at other times, as shown in the figure. This same behavior is observed for FSMs in the ‘stepB’ family with similar sizes. ROMs constructed for the ‘stepA’ model family were found to be more accurate in steady-state than those constructed for the ‘stepB’ family, perhaps due to the amount of energy captured by the states retained during model reduction. Simulation results for the FSM in the ‘stepB’ family with order $n = 25$ and input delay $nk = 2$, shown in the figure, show a 4 mm steady-state output error that is also generally less than 1 cm throughout the rest of the simulation. A similar observation was made about the output errors from simulation of the ROMs constructed from this model, where larger-order ROMs had larger errors, peaking between 2 and 3 cm. The errors in these simulations remain under 26.1% of the average duct height, well within the distance which the shock is expected to oscillate in steady-state.

Residual analysis of these simulations is shown in Figure 3.15, where the autocorrelations for both models families do not pass the whiteness test as well as the two previous simulations. For these simulations, the worst auto-correlation results for models in the ‘stepA’ family were observed for the order $n = 5$ model with input delay $nk = 1$. This contrasts to the results of simulation with the uniform-step input,
where this model has the best auto-correlation behavior. It was also observed that, for time delay values lower than that which the auto-correlation enters the CI, the auto-correlation between the FSMs and ROMs were more similar to each other than for the previous simulations. For models in the ‘stepB’ family, the auto-correlation calculations for models with input delay $nk = 1$ enter the CI for the smallest time delay, but, leave the CI at some larger delay. The only model in this family that has an auto-correlation that remains inside the CI ($nk = 0, n = 10$) also has an auto-correlation that enters the CI at the largest time delay. Cross-correlations of the FSMs are also similar to those of the ROMs, staying within the CI and behaving similarly to other simulations with validation inputs.
Figure 3.15: Residual analysis validation of the candidate model set against the ‘stepC’ data. The 99% confidence intervals (CI) are indicated by the shaded regions; the auto-correlation calculations had the same CI for each model, but the cross-correlation did not (indicated by boxes shaded with similar color to the line).

### 3.2.2 Constructing a candidate model set

Since the size of the isolator model is not assumed during control design, it is not necessary to identify a single model of the isolator from these results. Although extra effort will be required to implement the controller if the COM is a ROM, the control algorithm can also be developed independent of model size. Although 36 isolator models were constructed, validation revealed that many of the models, in particular some of the small-order ROMs, did not accurately capture the input-output behavior of the system. Analysis of the output errors in Tables 3.4 to 3.6, the residuals in Figures 3.11, 3.13, and 3.15, and the AIC in Figure 3.6 can be used to select a reasonably-sized set of the most accurate models.

Selecting first the models that have the lowest RMS error for the largest number of simulations, this set of models is reduced by one-third. Four FSMs were selected
from each model family: the four FSMs with delays $nk = 0$ and $nk = 1$ in the ‘stepA’ family, and the model of order $n = 20$ with input delay $nk = 0$, both models with delay $nk = 1$, and the model with delay $nk = 2$ in the ‘stepB’ family. The set was completed by selection of the most accurate ROMs: the order $k = 4$ ROM constructed from model of order $n = 15$ with delay $nk = 0$, and the order $k = 5$ ROMs from the order $n = 30$ models in the ‘stepA’ family, and the order $k = 3$ ROM of the model with order $n = 30$ and input delay $nk = 1$ in the ‘stepB’ family.

Residual analysis guides further reduction of this model set, where auto-correlation analysis suggests that models in the ‘stepB’ family do not have the same ‘whiteness’ as those in the ‘stepA’ family. The model in the ‘stepB’ family with the best auto-correlation result had input delay $nk = 2$ and order $n = 25$. Similarly, this analysis suggested that the model in the ‘stepA’ family with delay $nk = 2$ passes the whiteness test marginally better than the other models, but the RMS error analysis does not support the conclusion that this is one of the most-accurate models in the ‘stepA’ family.

For models with input delay $nk < 2$, the residual analysis provides conflicting conclusions: smaller-order FSMs have better auto-correlation when simulated with the ‘stepA’ and ‘stepB’ inputs, but the larger-order FSMs have better auto-correlations when simulated with the ‘stepC’ input. It is also difficult to use residual analysis for selecting the ROMs to include in the candidate model set, further necessitating the need to combine this analysis with RMS error analysis and AIC analysis. Doing so, three models from the ‘stepA’ family are chosen for the candidate model set. AIC calculations suggests that the most accurate model has order $n = 5$ and input delay $nk = 1$; the advantage of this model is that it is accurate as a FSM and small enough to be a COM. If the size of the FSM were not of importance, either model of order
n = 30 in the ‘stepA’ family may be chosen, as supported by RMS error and residual analysis, with an appropriate-sized ROM chosen for the COM.

If, in addition, a model from the ‘stepB’ family was to be considered, residual analysis and RMS error analysis suggests the model with input delay \( nk = 2 \) be chosen. In this case, the COM would be either the order \( k = 2 \) or \( k = 4 \) ROM constructed from this model. For comparison with the ‘stepA’ model family, the order \( n = 30 \) model with \( nk = 1 \) in the ‘stepB’ family may also be considered as a candidate model, even though this model is not as accurate as the other FSMs. The set of candidate models in Table 3.7 is representative of the full set of isolator models studied here and produces the most accurate results in simulation. The models in this set will be considered for control design, after first introducing uncertainty to each model.

3.3 Representing uncertainty in the model

Because a controller will be designed to anchor the shock wave in the presence of input (flow) perturbations, the control methodology will rely on robust control theory, which requires a representation of the model that contains uncertainty. The candidate isolators models in Table 3.7, constructed and validated in the previous section, represent nominal models of the isolator, which do not contain the modeling errors present in a mathematical model of a physical system. These errors are represented as uncertainty in the model (the plant).

Several possible representations of uncertainty are discussed by Zhou [46, 47]; the two main types are structured and unstructured uncertainty. Structured uncertainty manifests itself, for example, through parametric uncertainty in a model, while unstructured uncertainty represents a ‘disk’ in which the model exists. The latter representation is more useful in robust control theory, while the former is provided by
Table 3.7: Model order and input delay for each of four candidate models of the axisymmetric isolator, found using system identification and CFD data. Note that ‘stepA’ and ‘stepB’ refer to the set of data used for model identification.

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>nk</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepA</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>stepB</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2</td>
</tr>
</tbody>
</table>

MATLAB® during the system identification process. Here, structured uncertainty is transformed to an unstructured representation through one of three methodologies developed and applied to each candidate model in Table 3.7. In addition to introducing uncertainty to the models, application of these methodologies further reduces the size of the candidate model set for use in control design due to inaccuracies in the uncertainty representation for some models.

### 3.3.1 Structured Uncertainty Representation

When performing system identification, MATLAB® calculates a covariance matrix that contains information that provides the standard deviation of system parameters in various forms. For example, the matrices in the state equation of a system, in controllable canonical form, have the structure in equation (3.5), where the standard
deviations are calculated from the diagonal of the covariance matrix.

\[
A_{\sigma} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
a_1 \pm \sigma_{a_1} & a_2 \pm \sigma_{a_2} & a_3 \pm \sigma_{a_3} & a_4 \pm \sigma_{a_4} & a_5 \pm \sigma_{a_5} \\
\end{bmatrix}
= A \pm \sigma_A
\]

\[
B_{\sigma} = \begin{bmatrix}
b_1 \pm \sigma_{b_1} \\
b_2 \pm \sigma_{b_2} \\
b_3 \pm \sigma_{b_3} \\
b_4 \pm \sigma_{b_4} \\
b_5 \pm \sigma_{b_5} \\
\end{bmatrix}
= B \pm \sigma_B
\] (3.5)

MATLAB® provides several functions that use the covariance matrix to calculate the standard deviation of model parameters when expressed in other forms, like as a transfer function or through the frequency response.

In converting the model to a transfer function, the coefficients of the numerator and denominator of the nominal transfer function are calculated from the matrices \(A, B, C,\) and \(D\) and the standard deviations of these coefficients are calculated from the \(a_i\) and \(b_i\) in equation (3.5). The structure of the uncertain transfer function is shown in equation (3.6), where the standard deviations of the numerator coefficients, \(c_i,\) are functions of \(a_i\) and \(b_i:\)

\[
P_{\sigma}(s) = \frac{(c_1 \pm \sigma_{c_1})s^4 + (c_2 \pm \sigma_{c_2})s^3 + (c_3 \pm \sigma_{c_3})s^2 + (c_4 \pm \sigma_{c_4})s + (c_5 \pm \sigma_{c_5})}{s^5 - (a_5 \pm \sigma_{a_5})s^4 - (a_4 \pm \sigma_{a_4})s^3 - (a_3 \pm \sigma_{a_3})s^2 - (a_2 \pm \sigma_{a_2})s - (a_1 \pm \sigma_{a_1})}
\]

\[
= \frac{P_{n}(s) \pm \sigma_{P,n}(s)}{P_{d}(s) \pm \sigma_{P,d}(s)}
\] (3.6)

Models with the forms in equations (3.5) and (3.6) will be referred to in the following discussion as ‘parametrically uncertain’ models, to differentiate them from models
with unstructured uncertainty (‘uncertain models’). The first methodology developed here for transforming structured to unstructured uncertainty considers parametrically uncertain models of the isolator in both state-space and transfer function forms.

The magnitude and phase data of the frequency response of the model can be shown in a Bode plot and MATLAB® contains a function that calculates the standard deviation of the frequency response data from the information in the covariance matrix. Then the bounds on the frequency response of the uncertain model \( P_\Delta \) are: 

\[
|P_\Delta| = |P| \pm \sigma_m \quad \text{and} \quad \angle P_\Delta = \angle P \pm \sigma_p.
\]

The second methodology presented here for transforming the uncertainty representation constructs the uncertain model from the magnitude data of the frequency response.

A third methodology is then presented that implements the first two methodologies using a built-in MATLAB® function that directly constructs a model with multiplicative uncertainty. The models constructed through this methodology contain more conservative estimates of the uncertainty than those constructed from the first two methodologies.

### 3.3.2 Unstructured Uncertainty Representation

In Refs. [46, 47], Zhou introduces several representations of unstructured uncertainty in a multi-input multi-output (MIMO) model, where the location of the uncertainty affects the structure of the uncertain models. For SISO models like the isolator models considered here, some of the representations are equivalent, such as the ‘input multiplicative’ and ‘output multiplicative’ representations [46, 47]. Here, both additive and multiplicative uncertainty will be considered when possible; additive uncertainty is represented with the form \( P_\Delta = P + W_\Delta \) and multiplicative uncertainty with the form \( P_\Delta = P(1 + W_\Delta) \). In these systems, the function \( W \) is a
weighting function that ‘shapes’ the uncertainty, $\Delta$, which is here assumed to satisfy $|\Delta| < 1$.

As mentioned briefly in Section 3.3.1, two distinct methodologies for constructing an uncertain model will be presented: one using the parametrically-uncertain model structures (in equations (3.5) and (3.6)) and one using frequency response data. A third methodology will then be developed that uses a built-in MATLAB® function to construct the uncertain models; the results of this methodology will be presented in Section 3.3.3 along with the other results for comparison purposes.

**Method 1: Transforming from Parametric Uncertainty**

The first methodology considers the parametrically-uncertain models in state-space and transfer function forms, where uncertainty enters the model as shown in equations (3.5) and (3.6), respectively. This analysis considers forms of the model in both the discrete time (DT) and continuous time (CT) domains, where the bilinear transform is used to convert between time domains. The nominal models constructed in Section 3.1.2 are in DT, since sampled data was used during system identification. In discussion of this methodology, and the systems that result, models constructed from the state-space form will be referred to as ‘state-space-based models’ and those constructed from the transfer function form will be ‘transfer function-based models.’

As discussed in Section 3.3.1, arrays of transfer function coefficients and their standard deviations can be calculated using the covariance matrix. Since MATLAB® does not contain a function to transform models of these forms directly to uncertain models, it was necessary to create two functions, which take data from the nominal model to construct either an uncertain state-space model or arrays of uncertain transfer function coefficients.
Once the parametrically-uncertain models have been constructed as objects that can be used by functions in the Robust Control Toolbox (RCT), construction of models with unstructured uncertainty can proceed. One feature of the RCT is the ability to uniformly sample the uncertain parameters across the range of uncertainty to construct several models of the system [49,50]. The state-space models and the transfer function coefficients are both sampled to produce two sets of models in the DT domain. Each models in these sets will be referred to as ‘sampled models’ in the following discussion.

Because the canonical form of a model is often ill-conditioned, the sampled state-space models are balanced before continuing with analysis. Each model is then converted from DT to CT, doubling the number of state-space-based models under consideration. Similarly, each pair of sampled numerator and denominator polynomials are used to construct a DT transfer-function model, which is also converted to CT, doubling the number of transfer function-based models.

Given the nominal DT and CT models \( P \), models of the ‘deviation’ \( W\Delta \) can be constructed for each sampled models \( P_\Delta \). For each model, the additive uncertainty deviation \((W\Delta)_{\text{add}}\) was found by calculating the difference between the sampled and nominal models, \((W\Delta)_{\text{add}} = P_\Delta - P\), and the multiplicative uncertainty deviation was found as the relative difference between the models, calculated with respect to the nominal model, \((W\Delta)_{\text{mult}} = (P_\Delta - P)/P = P_\Delta/P - 1\).

A search was implemented to determine the maximum magnitude of each \( W\Delta \), assumed to be \( |W| \) since \( |\Delta| \leq 1 \). Each set of data, represented by pairs of frequency \( \omega \) and magnitude \( |W| \), was fit to a system \( W_{\text{fit}} \), the weighting function in the uncertain model. Each of these weighting functions was a state-space model, constructed with the same order as the nominal model \( P \), that will be shown in Section 3.3.3 to, in general, fit the data well. Given these fits, two uncertain models could be
defined: \( P_{\Delta,A} = P + W_{\text{fit},A}\Delta \) and \( P_{\Delta,M} = P(1 + W_{\text{fit},M}\Delta) \), where \( \Delta \) is a linear system with uncertain dynamics bounded by \( |\Delta| \leq 1 \). Comparison of these models to the parametrically-uncertain models from which they were constructed, and to results from other methodologies, will be detailed in Section 3.3.3.

**Method 2: Using Frequency Response Data**

The second methodology uses the frequency response data to construct an uncertain model of the isolator, where the standard deviations of both the magnitude and phase are determined from the covariance matrix of the DBM. Unlike method 1, this methodology is only implemented for the state-space model in discrete time, as covariance data is not preserved through conversions between object types or time domains in MATLAB\textsuperscript{®}. In addition, only a model with multiplicative uncertainty is considered, due to the available data and the necessity of implementing two fits for a model with additive uncertainty.

Because analysis of model robustness generally concerns bounds on the magnitude of signals or systems, the magnitude of the frequency response is of most interest. Considering the frequency response of the uncertain model \( P_{\Delta} \) as a phasor, the magnitude of the frequency response is the product of the magnitude of the nominal model \( P \) and the magnitude of the deviation, as shown in equation (3.7):

\[
\begin{align*}
P_{\Delta} &= P(1 + W_{\Delta}) \\
|P_{\Delta}| &= |P(1 + W_{\Delta})| \\
&= |P||1 + W_{\Delta}|
\end{align*}
\]
From the frequency response data, the relationship in equation (3.8) can be determined as

\[
|P| + \sigma_m = |P| \left(1 + \frac{\sigma_m}{|P|}\right)
\]

\[
1 + \frac{\sigma_m}{|P|} > 0 \Rightarrow \left|1 + \frac{\sigma_m}{|P|}\right| = 1 + \frac{\sigma_m}{|P|} = |1 + W\Delta|
\]

The final line in equation (3.8) relates the magnitude response of the uncertain model \(P_\Delta\) to the magnitude response of the DBM; given this relation, method 2 employs a similar strategy as method 1 to construct the weighting function \(W\) of the uncertain model.

A model of an intermediate system is fit to the frequency response data \(1 + \sigma_m/|P|\), which has been calculated point-by-point for each model. The intermediate model, \(P_{int}\), has the same number of states as the nominal model. Since \(|P_{int}| = |1 + W\Delta|\), it can be assumed that \(P_{int} = 1 + W\Delta\), from which \(W\) can be calculated under the assumption that \(|\Delta| = 1\): \(W = P_{int} - 1\). The uncertain model \(P_{\Delta,2} = P(1 + W\Delta)\) can then be constructed using this weighting function. For this model, the uncertainty is characterized as \(\Delta \in \mathbb{C}\) with \(|\Delta| < 1\) to produce an uncertain model which covers the range of uncertainty more completely than was found for \(\Delta\) characterized as an uncertain linear model. Further analysis of this methodology will be given in Section 3.3.3.

**Method 3: Implementing methods 1 and 2 through built-in MATLAB® functions**

Included in the RCT is a function which produces results similar to those achieved from methods 1 and 2. This function takes as inputs a nominal model, a set of models representing the uncertainty in the system, and a model order to produce a model with multiplicative uncertainty and a weighting function that ‘covers’ the uncertainty. The
way this function is implemented allows for methods 1 and 2 to be re-implemented and for a comparison of the models produced by the three methodologies.

To implement method 1, the set of representative models is constructed by sampling the uncertain state-space model or uncertain transfer function coefficients. The size of this set can be increased, if desired, to improve the accuracy of the uncertain model. In contrast, implementation of method 2 is limited to include just two representative models of each nominal model, which have magnitude responses $|P| \pm \sigma_m$.

One other difference between the implementations is that, for method 1, the set of representative systems can be defined as either an array of state-space models or as an array of frequency-magnitude pairs, whereas for method 2 only the latter form can be used without the need to run an auxiliary fit.

The results of applying this methodology will be presented alongside those from methods 1 and 2 in the next section.

### 3.3.3 Verifying Unstructured Representation

Each methodology was implemented on the four candidate models in Table 3.7. Rather than validating the models, it is instead verified that the weighting functions produce uncertain models that covers the set of parametrically-uncertain models (method 1) or the range of frequency responses (method 2).

In the following results, models with additive uncertainty are evaluated only for method 1, as the other methodologies produced only models with multiplicative uncertainty. Model-by-model results are presented for methods 1 and 3, as four different forms of each candidate model are analyzed. Along with separate analysis of the fits resulting from implementation of each methodology, the results will be compared to each other in order to draw final conclusions.
Verification of Method 1

The results of this relatively complex method will be presented in three parts: examining uncertainty in the model structure, comparing the fits \( W_{\text{fit},M} \) to the deviation systems \( (W \Delta)_{\text{mult}} \) and comparing the ‘upper-bounds’ of the uncertain systems to a set of parametrically-uncertain models different from those used to find \( W_{\text{fit}} \). Results for the system with additive uncertainty are only considered in this final step of analysis, where the uncertain models with the form \( P + W_{\text{fit},A} \Delta \) are shown to be inaccurate.

**Effect of Uncertainty on Matrix Elements.** Before proceeding with verification of method 1, it is of interest to study how the elements of each balanced system matrix vary in comparison to the nominal values. By construction, the parameters in the last row of \( A \) and every element in \( B \), when expressed in canonical form, vary symmetrically about the nominal value, while the \( C \) matrix is fixed and \( D = 0 \). It will be seen that the transformation to balanced coordinates affects the presence of uncertainty in the elements of the \( A, B \) and \( C \) matrices.

Considering the order \( n = 5 \) model in the candidate set, let the elements of \( A \) be numbered left-to-right, top-to-bottom, so that the element in the first column of the first row is labeled 1, the element in the fifth column of the first row is labeled 5, and so on. In Figure 3.16, the elements of \( A \) are plotted along an \( x \)-axis with coordiantes corresponding to these assigned labels. In the figure, the nominal value of each element is indicated by a star, and the circles represent the value of that same element for each sampled DT system.

It can be seen that some elements, most notably \((1,1), (2,5), (3,3), (4,2), \) and \((5,2)\), have a more symmetrical distribution about the nominal value than others. The nominal value of element \((3,2)\) appears to be a lower bound while, for element \((4,1)\), it is an upper bound. Other elements, such as \((1,5)\) and \((2,2)\), vary in an
Figure 3.16: Elements of the $A$-matrix for the nominal (star) and sampled (circle) $n = 5$ models, in balanced coordinates. The variation due to uncertainty is clearly seen. (The elements are numbered with an index that increases across the rows of the matrix.)

asymmetric fashion about the nominal value. Two elements ((1, 4) and (5, 5)) have very little variation.

Like the elements of the $A$ matrix, those of the $B$ matrix in Figure 3.17 are not symmetrically distributed about the nominal value like they are in the canonical representation of the matrix. Here, each element varies more toward values below the nominal value, with the first element varying only in this direction and the variation of the third element showing the most symmetry.

As mentioned previously, in canonical form the $C$ matrix is a fixed matrix (for this system, $C = [1 \ 0 \ 0 \ 0 \ 0]$) but becomes an uncertain matrix through the transformation to balanced coordinates, as can be seen in Figure 3.18. The variation of each element is similar to that observed in the previous two matrices and is not generally symmetrical about the nominal value.
Figure 3.17: Elements of the $B$-matrix for nominal (star) and sampled (circle) $n = 5$ models, in balanced coordinates, showing variation due to uncertainty.

Figure 3.18: Elements of the $C$-matrix for nominal (star) and sampled (circle) $n = 5$ models, in balanced coordinates. Note that the $C$ matrix is certain in canonical coordinates, but becomes uncertain under a coordinate transformation.
If the number of models under consideration were increased, it may be possible to construct a representation of uncertainty from the information in Figures 3.16 to 3.18. In the present case, however, these figures are more useful in showing uncertainty in the state-space model of the isolator rather than acting as part of a methodology for constructing an uncertain isolator model.

Evaluating the Fit for $W$. The fit of the weighting function $W$ calculated following method 1 is evaluated first. Only the results for models with multiplicative uncertainty are presented here, as the fits for models with additive uncertainty were found to be of similar accuracy. One figure is presented for each candidate model, and each subplot shows the frequency responses of the $(W\Delta)_{\text{mult}}$ constructed for each model form in each domain. For state-space-based models in DT, the frequency responses are shown in the top-left subplot, and the responses in the bottom-left subplot are for CT state-space-based models. Similarly, frequency responses for DT transfer function-based models are in the top-right subplot, and those for CT transfer function-based models are in the bottom-right. The blue solid line shows the point-wise maximum of the magnitude responses and the frequency response of the fitted system is shown by the dashed lines.

Figure 3.19 shows the frequency response for the smallest-order model ($n = 5$), where each weighting function can be seen to accurately fit the magnitude data. It appears that the fits for the transfer function-based models, particularly in CT, are more accurate than for the state-space-based models, but in all cases $W_{\text{fit},M}$ is a good bound on $(W\Delta)_{\text{mult}}$.

When the order of the ‘stepA’ family model increases to $n = 30$, one of the first observations to be made concerns the high-frequency response of models in the DT domain. As can be seen in Figure 3.20, the frequency response is less-smooth in certain frequency ranges than that of the order $n = 5$ model and suggests the potential
Figure 3.19: Magnitude responses of $(W\Delta)_{\text{mult}}$ for the $n = 5, nk = 1$ model in the ‘stepA’ family, with the maximum value and fit for $W$ indicated as the solid and dashed lines, respectively.

for a loss of accuracy when fitting the weighting function to the data. (In the CT domain, similar behavior is seen in the middle of the frequency range due to transformation between time domains.) In particular, for this model, each fit underestimates $(W\Delta)_{\text{mult}}$ in the low-frequency range, due to the need for the additional states in $W$ to capture high-frequency dynamics. This was partially rectified by defining a lower bound on the fit for $W_{\text{fit,M}}$ to construct a weighting function with a more conservative fit. Since the frequency response of these models is more conservative than those shown in Figure 3.20, but less conservative that the results from implementation of method 3, they are not shown here.

Application of method 1 to the $n = 30$ model in the ‘stepB’ family, shown in Figure 3.21, produces a more accurate model than found for the ‘stepA’ family with the same order. The fits are more accurate for models in the CT domain, although in
the low-frequency range the fits constructed for DT models are inaccurate, but more conservative than the results for the equivalent model in the ‘stepA’ family.

The extra relative degree in the $n = 25$ model does not have much of an effect on the fits for $(W\Delta)_{\text{mult}}$, as seen in Figure 3.22. As with the $n = 30$ model, the fits overestimate the low-frequency bounds in state-space-based models and underestimate these bounds for transfer function-based models, an inaccuracy likely due to the high-frequency variation in the frequency responses.

Evaluating the Bounds on Parametrically-Uncertain Models. Once the fits for $W\Delta$ have been evaluated, the weighting functions $W_{\text{fit,A}}$ and $W_{\text{fit,M}}$ are used to construct two models: $P_{\text{up,A}} = P + W_{\text{fit,A}}$ and $P_{\text{up,M}} = P(1 + W_{\text{fit,M}})$. Since $P_{\text{up}}$ is constructed under the assumption $\Delta = 1$, these two models can be considered as the upper-bound on the uncertain model, and will be referred to as such in the following discussion. In order to verify the accuracy of these upper-bound models,
Figure 3.21: Magnitude responses of $(W\Delta)_{\text{mult}}$ for the $n = 30$, $nk = 1$ model in the ‘stepB’ family, with the maximum value and fit for $W$ indicated as the solid and dashed lines, respectively.

Figure 3.22: Magnitude responses of $(W\Delta)_{\text{mult}}$ for the $n = 25$, $nk = 2$ model in the ‘stepB’ family, with the maximum value and fit for $W$ indicated as the solid and dashed lines, respectively.
the frequency response of each is plotted with the frequency responses of the nominal model and those of a set of parametrically-uncertain models different from those from which $W_{\text{fit}}$ was constructed.

The subplots in each figure are arranged the same way as those presented in discussion of the accuracy of the fits for $W$. In each subplot, the frequency responses of each parametrically-uncertain model are shown by dashed lines and the nominal frequency response by a solid line. The frequency response of the upper-bound on the model with additive uncertainty, $P_{\text{up, A}}$, is shown as a dotted magenta line and the dotted cyan line is the magnitude of $P_{\text{up, M}}$, the upper-bound on the model with multiplicative uncertainty.

Figure 3.23 shows that the weighting function $W_{\text{fit, M}}$ produces a model that accurately represents the upper bound of the uncertainty in the $n = 5$ model, while the weighting function $W_{\text{fit, A}}$ does not. The fit for the model with multiplicative uncertainty may be further improved by using a only set of stable systems in calculating the fit for $W_{\text{fit, M}}$. (This check, however, can only be implementing for the small-order model and are therefore not included here.)

Note that, in all cases, the upper-bound on the model with multiplicative uncertainty suggests that uncertainty is concentrated at lower frequencies rather than at higher frequencies. The opposite was found for the models constructed through method 2, where uncertainty in the DT models was more concentrated at higher frequencies. It is in the frequency ranges where uncertainty is small that models with additive uncertainty fail to produce a useful upper bound.

As with the fits for $W$, additional drawbacks are observed when the order is increased. For the $n = 30$ model in the ‘stepA’ family, the type of model affects how these present themselves. When the model is state-space-based, the bound constructed for the model with multiplicative uncertainty is more accurate than that
constructed for the model with additive uncertainty, as can be seen in Figure 3.24.

For the CT model, the bound from the model with additive uncertainty is more accurate than in DT, where the worst estimate of the upper bound occurs in the frequency range slightly above that where the variation in the magnitude response is large.

The results for the CT transfer function-based model are similar to those of the CT state-space-based model, and the results for the transfer function-based model represented in the DT domain are the poorest. Although accurate at high frequencies, the upper bounds of the models, with both additive and multiplicative uncertainty in this form and time domain, are below the nominal value at low frequencies even though some parametrically-uncertain models have a magnitude larger than the nominal system at these frequencies.

The $n = 30$ model in the ‘stepB’ family has poor results for both DT models, as can be seen in the plots in the top row of Figure 3.25, where the upper bounds have
The results for the $n = 25$ model in the ‘stepB’ family, given in Figure 3.26, are similar for three of the four models (the two in the CT domain, and the state-space-based model in DT). The upper bound on the DT transfer function-based model with multiplicative uncertainty greatly overestimates uncertainty at low frequencies, in contrast to the results for the $n = 30$ model, and can be attributed to the set of parametrically-uncertain models used to construct the weighting function $W_{\text{fit},M}$ for this model.
Bode Plots of SS–based (L) and TF–based (R) Models in DT (T) and CT (B)
(stepB, \(n = 30\), \(nk = 1\))

Figure 3.25: Frequency responses of the \(n = 30\), \(nk = 1\) model in the ‘stepB’ family, with the nominal model (solid), additive (magenta dots), and multiplicative (cyan dots) upper bounds indicated.

Bode Plots of SS–based (L) and TF–based (R) Models in DT (T) and CT (B)
(stepB, \(n = 25\), \(nk = 2\))

Figure 3.26: Frequency responses of the \(n = 25\), \(nk = 2\) model in the ‘stepB’ family, with the nominal model (solid), additive (magenta dots), and multiplicative (cyan dots) upper bounds indicated.
It is not possible to implement a check of stability on the sampled models with order \( n = 25 \) or \( n = 30 \) due to high sensitivity between stability and uncertainty, precluding the ability to improve the results in Figures 3.24 to 3.26 through exclusion of systems with unstable dynamics.

**Verification of Method 2**

The results obtained through implementation of method 2 on the smallest-order model \( (n = 5) \) were very accurate, but when implemented on the other candidate models in Table 3.7, the results proved to be disappointing, as Figure 3.27 shows. The frequency responses for each model are shown in separate subplots, with the ‘stepA’ models shown in the first row and ‘stepB’ models in the second row and the corresponding \((n, nk)\) pairs given by the order in the table. In each plot, the frequency response of the nominal model is shown by the green line and the red line shows the frequency response of the upper bound models calculated from the fit \( W \), similar to \( P_{up,M} \) introduced in discussion of the results from method 1.

A distinction should be made between the definition of \( \Delta \) for models constructed following method 1 and those constructed following method 2. In this case, \( \Delta \) is a complex gain with the bound \(|\Delta| \leq 1\), while \( \Delta \) is an uncertain linear system with gain less than 1 for each of the other uncertain models considered. When \( \Delta \) is defined as an uncertain linear system for models constructed from method 2, the resulting systems do not cover the range of uncertainty in the frequency response data calculated from the DBM and covariance matrix.

It can be seen in Figure 3.27 that, for the smallest-order model \((n = 5)\), uncertainty is concentrated at high frequencies and the frequency response of \( P(1 + W) \) produced an accurate approximation of \(|P| + \sigma_m\). In the larger-order models, uncertainty is more concentrated at low frequencies, as was found for uncertain systems.
Figure 3.27: Frequency responses of models with multiplicative uncertainty determined through implementation of method 2; the nominal system is indicated by the thick green line and the upper bound $P(1 + W)$ by the thick red line. The models are (left to right, top to bottom): $n = 5, nk = 1$, ‘stepA’; $n = 30, nk = 1$, ‘stepA’; $n = 30, nk = 1$, ‘stepB’; $n = 25, nk = 2$, ‘stepB’.

constructed through implementation of method 1 and the upper bound did not accurately approximate $|P| + \sigma_m$ in these ranges. This inaccuracy may be attributed to the large variations at high frequencies introduced by additional states in the model.

Verification of Method 3

Although method 3 was implemented using the formulations of both methods 1 and 2, the results presented here, in Figures 3.28 to 3.31, focus on the relationship between method 3 and method 1, since they are directly comparable to Figures 3.19 to 3.22. The subplots are arranged in the same way as the results for method 1 and show the frequency responses of $(W\Delta)_{\text{mult}}$ and the magnitude of the weighting function $W_1$ calculated through the built-in MATLAB® function.
Figure 3.28: Magnitude repsonses of \((W\Delta)_{\text{mult}}\) used to construct the weighting function \(W_1\) (dashes) in the uncertain ‘stepA’ model with \(n = 5, nk = 1\), using \textit{ucover}.

The fits for the smallest-order model, shown in Figure 3.28, are most accurate in estimating the upper bound on \((W\Delta)_{\text{mult}}\) for DT models. In the CT domain, the weighting function underestimates the magnitude at low frequencies, while otherwise producing a good fit, similar to what was observed for direct implementation of method 1.

The same general behavior of \(W\) can be observed for the higher-order models in Figures 3.29 to 3.31. In general, the weighting function calculated by MATLAB\textsuperscript{®} accurately estimates the upper bound across the entire frequency range for DT models, but underestimates the bound in the low-frequency range for CT models. It will be shown that this bound is more conservative than that calculated through direct implementation of method 1.
Figure 3.29: Magnitude responses of \((W\Delta)_{\text{mult}}\) used to construct the weighting function \(W_1\) (dashes) in the uncertain ‘stepA’ model with \(n = 30, nk = 1\), using \text{ucover}.

Figure 3.30: Magnitude responses of \((W\Delta)_{\text{mult}}\) used to construct the weighting function \(W_1\) (dashes) in the uncertain ‘stepB’ model with \(n = 30, nk = 1\), using \text{ucover}.
Comparison and Summary of Results

The uncertain models constructed through implementation of each of the three methodologies detailed in Section 3.3.2 is presented here in three parts. First, the frequency responses of $P_{\text{up, } M}$ calculated for each method are plotted with those of a set of parametrically-uncertain models, allowing for comparison of the bounds to the uncertain model set (most applicable to method 1). Then the frequency responses of the weighting functions are compared, followed by comparison of the frequency responses of each uncertain model, allowing for a comparison of the ‘range of uncertainty’ covered by each model.

Comparison of Upper Bounds with Parametrically-Uncertain Systems. It is important to verify that $P_{\text{up, } M} = P(1+W)$ produces an acceptable approximation of the uncertainty in the parametrically-uncertain models used to construct $W$. In addition, these upper bounds should be compared to each other to evaluate how
conservative the model is in approximating uncertainty, which will drive the choice of method used to represent uncertainty in each model.

A figure is presented for each candidate model in Table 3.7, where the subplots are arranged similarly to previous figures: DT models in the top row, CT models in the bottom row; state-space-based models in the left column, transfer function-based models in the right column. The dashed lines show frequency responses for a set of sampled parametrically-uncertain models and the nominal frequency response is shown in orange. The frequency responses of the upper bounds produced by implementing method 1 are shown in red for the direct methodology and in black for the implementation through method 3. Similarly, the plots in cyan and magenta show frequency responses for implementation of method 2 directly and through method 3, respectively. Note that method 2 was only implemented on the DT state-space model.

Figure 3.32 compares the bounds for the small-order model with \( n = 5 \). Implementation of method 3 generally produced a more conservative bound in relation to direct implementation of either method, although some deviation in this trend is observed for the CT state-space model. For this nominal model only, application of method 2 through method 3 produced bounds that are more similar to the results of directly applying method 1 rather than method 2.

The location of uncertainty in the frequency domain can also be observed. Although difficult to see in Figure 3.32, only direct implementation of method 2 produces a model where uncertainty is concentrated at high frequencies. In all other cases, uncertainty is distributed across the entire frequency range, and most concentrated on the low end of the range (for DT models). While it is possible this is the case for the isolator model, it is typically expected that uncertainty is concentrated at higher frequencies since DBMs do not generally contain enough states to capture
Figure 3.32: Comparison of the uncertain model with multiplicative upper bounds, $P_\Delta = P(1 + W)$, for application of the methodologies considered in Section 3.3.2 to the ‘stepA’ model with $n = 5, \ nk = 1$. The plot includes frequency responses of parametrically-uncertain systems not used to find $W$.

these dynamics. This suggests that method 2 may be the best choice for representing uncertainty in this particular model.

The general behavior of the frequency responses of these upper-bound models does not change when the order of the model is increased, as Figure 3.33 shows for the $n = 30$ model in the ‘stepA’ family. Like the $n = 5$ model, the weighting function constructed through implementation of method 1 through method 3 is a more conservative estimation of uncertainty than that constructed through direct implementation of method 1. This trend holds even in the high-frequency ranges, where the frequency response shows large variations in magnitude.

Implementation of method 2 to this model produces bounds that are very different from those constructed for the smaller-order model. It was seen that direct application of method 2 to the order $n = 5$ model resulted in an uncertain model with a
frequency response that was an excellent fit to the one-standard deviation bound on
the frequency response of the DBM. Implementation of this method through method
3 produced a much more conservative bound, but not as conservative as the bounds
constructed through implementation of method 1. For the \( n = 30 \) model, however,
the bound resulting from application of method 2 through method 3 was similar to
that from implementing method 1 through method 3 at low frequencies, and to that
from direct implementation of method 1 at high frequencies. Direct application of
method 2, as previously discussed, produced a bound that was far more conservative
than the other bounds at low frequencies, though similar at high frequencies. And,
in comparison to the uncertain frequency response of the DBM, the fit is not as ac-
curate as for the smaller model. This suggests that method 1 may be a better choice
for representing uncertainty in this model, although uncertainty is concentrated in a
different frequency range than expected.

For the two models in the ‘stepB’ family, with frequency responses given in Fig-
ures 3.34 and 3.35, the results for application of method 1 follow the general trend
observed for the other models, where application of method 1 through method 3
produced a more conservative approximation of the upper bound than direct imple-
mentation. For these models, the uncertainty estimations differ more from each other
than those observed for the ‘stepA’ family of models, especially the \( n = 30 \) model.

Comparison of the bounds resulting from both implementations of method 2 re-
veals some noticeable differences. In particular, the results for implementing the
methodology through method 3 are more comparable to those from direct implemen-
tation of method 1. Like the \( n = 30 \) model in the ‘stepA’ family, direct implementa-
tion of method 2 yields the most conservative approximation at lower frequencies
for the \( n = 25 \) model, but the least conservative estimate at high frequencies. For
the \( n = 30 \) model in the ‘stepB’ family, however, the results of direct application
Nominal and Upper-bound Responses with Parametrically-Uncertain Models (stepA, n = 30, nk = 1)

Figure 3.33: Comparison of the uncertain model with multiplicative upper bounds, $P_\Delta = P(1 + W)$, for application of each of the methodologies considered in Section 3.3.2 to the ‘stepA’ model with $n = 30$, $nk = 1$. The plot includes frequency responses of parametrically-uncertain systems not used to find $W$.

of method 2 are only slightly more conservative than those from applying method 1 through method 3. In all cases, the uncertainty is concentrated more in the low frequency range, as with the $n = 30$ model in the ‘stepA’ family.

Comparison of Weighting Functions. The weighting function $W$, constructed through implementation of each methodology to each isolator model is used to define the frequency response of the upper-bound model $P_{up,M}$. Although the frequency responses of the weighting functions in Figures 3.36 and 3.37 do not reveal additional insight on the uncertain models, they act as another point of comparison.

Each figure shows the frequency responses of the weighting functions for models in a specified time domain, where DT weighting functions are shown in Figure 3.36 and CT weighting functions are shown in Figure 3.37. The subplots are arranged like those in the figures presented in discussing the results for method 2, where the
Figure 3.34: Comparison of the uncertain models with multiplicative upper bounds, $p_\Delta = P(1 + W)$, for application of each of the methodologies considered in Section 3.3.2 to the ‘stepB’ model with $n = 30$, $nk = 1$. The plot includes frequency responses of parametrically-uncertain systems not used to find $W$.

Figure 3.35: Comparison of the uncertain models with multiplicative upper bounds, $P_\Delta = P(1 + W)$, for application of each of the methodologies considered in Section 3.3.2 to the ‘stepB’ model with $n = 25$, $nk = 2$. The plot includes frequency responses of parametrically-uncertain systems not used to find $W$. 
frequency responses of models in the ‘stepA’ family are in the first row, and those from ‘stepB’ family models are in the second row.

Considering first the results for DT models, the source of some discrepancies between the frequency responses of uncertain ‘stepA’ family models constructed through implementation of method 2, figures 3.32 and 3.33, can immediately be observed. Recall that, for the order \( n = 5 \) model, \( P_{\text{up},M} \) had a frequency response that closely approximated the uncertain frequency response of the nominal model (uncertainty concentrated at high frequencies). In contrast, the approximation of the upper bound of the order \( n = 30 \) model was more conservative at low frequencies. The differing frequency responses are related to the respective weighting functions, which have the frequency responses shown in Figure 3.36 and are much different from the other weighting functions constructed for the DT models. The magnitude of the weighting function constructed for the order \( n = 5 \) model using method 2 is less than 1 at all frequencies, while the magnitudes of the other weighting functions constructed for this model are above 1 for the entire frequency range, except for a small high-frequency region. Similarly, at low frequencies the weighting function constructed using method 2 with the order \( n = 30 \) model is below 1 until the frequency at which the magnitude of the model begins to vary rapidly. For each of the other models, and for all of the other methodologies, the magnitude of the weighting functions is generally greater than 1. The magnitude of \( W \) describes the relative uncertainty at each frequency, so that in ranges where \( |W| << 1 \), there is not much uncertainty in the model and as \( |W| \) increases above 1, uncertainty in the model similarly increases. Considering the model with order \( n = 5 \), the weighting function constructed using method 2 suggests a strong concentration of uncertainty at high-frequencies, while each of the other weighting functions suggest a model with uncertainty present throughout the frequency spectrum, with some concentration at low and high frequencies. The
shapes of the frequency responses of $W$ are reflected in the concentration of uncertainty in the models with multiplicative uncertainty constructed from each weighting function.

A similar trend can be observed from the frequency responses of the weighting functions constructed for defining uncertain models in CT, shown in Figure 3.37: each weighting function has a magnitude that, in general, is above 1 except for a small range where the magnitude is between 0.1 and 1. As observed for the DT models, the weighting functions are of similar magnitude except in some small frequency ranges, which affects how conservative the bounds on uncertainty in the model are.

**Comparison of Uncertain Model Frequency Responses.** Uncertain models with the structure $P_\Delta = P(1 + W\Delta)$ are constructed for each weighting function $W$ shown in Figures 3.36 and 3.37. The uncertainty $\Delta$ is a linear system with uncertain
Figure 3.37: Comparison of $|W|$ found from application of each methodology to construct a continuous-time model with multiplicative uncertainty.

Each uncertain model is then ‘sampled’ to obtain a set of five sampled models, and the frequency responses of each sampled model are compared to the frequency response of the nominal model, and to each other. These comparisons are shown in Figures 3.38 and 3.39, which are organized in a similar way to Figures 3.36 and 3.37. The frequency responses of the sampled models are shown as thin solid lines in each subplot, and the nominal frequency response is included for reference. It was verified that, as expected, each nominal model had the same frequency response, providing a check that implementation of methods 1, 2, and 3 preserved the nominal model.

The results in Figure 3.38 suggest that the small-order ($n = 5$) DT model should be considered separately from the other models under consideration. It was observed
that, as expected, sampled models from the uncertain order \( n = 5 \) model constructed through direct implementation of method 2 had different frequency responses than the other sampled models. These frequency responses also reflect the fact that uncertain models constructed through method 3 had a more conservative upper bound, as the magnitudes of those frequency responses were generally larger than those of the models sampled from the related uncertain model constructed through direct implementation of method 1. In addition, these plots show that the frequency responses of models constructed through implementation of method 2 through method 3 are similar to those constructed through implementation of method 1.

In general, these observations hold for the larger-order candidate models, where the frequency responses of models results from direct implementation of method 1 and from implementation of method 2 through method 3 are similar. For the large-order models, however, the frequency responses of models sampled from the uncertain model constructed through direct application of method 2 are easily distinguished from the frequency responses of the other models. This observation reinforces the idea that method 2 is not an acceptable choice for representing uncertainty in large-order DT models.

Unlike the DT models, the frequency responses of the sampled CT models are similar to each other. In general, the amount of uncertainty is reflective of how conservative the weighting function was constructed to be: models resulting from applying method 3 capture more uncertainty, but still have frequency responses close to those of the other models. This leaves the choice of uncertain model to be made based on how conservative the desired approximation of uncertainty should be.

One final consideration in choosing an uncertain model representation is the stability of the sampled models. Each model constructed through implementation of method 3 is guaranteed stable by the built-in MATLAB® function, so it remained
Figure 3.38: Comparison of the frequency responses of each uncertain discrete-time model constructed as described in Section 3.3.2.

Figure 3.39: Comparison of the frequency responses of each uncertain continuous-time model constructed as described in Section 3.3.2.
to verify the stability of the models constructed through direct implementation of methods 1 and 2. Each uncertain model was sampled and the stability of each sampled model was checked; it was found that method 1 produced an uncertain model with stable sampled models for each candidate model, while the only uncertain model constructed through implementation of method 2 that produced stable sampled models had order $n = 5$. This further enforces the conclusions that method 2 can only be applied successfully to the smallest candidate model, while method 1 is the best choice for representing uncertainty in the other models.

### 3.3.4 Refining the candidate model set

Efforts for representing uncertainty in the candidate isolator models yielded a set of models with multiplicative uncertainty, where the uncertainty representations were constructed from parametric uncertainty in the data-based isolator models. Examination of the results reveals that, instead of applying the same methodology to each candidate model, or a separate methodology to each model, the candidate model set should be refined through identification of a single isolator model. The results in Section 3.3.3 suggest that the order $n = 5$ model is the best choice since the uncertain model constructed through implementation of method 2 has uncertainty concentrated at high frequencies, as expected. This model also has the advantage of being small enough for control design without the need to reduce the model and propagate uncertainty to a ROM.

This choice is further supported by results found during system identification. Validation of the models in Section 3.2 and calculation of the AIC and the empirical best fit parameter in Section 3.1.2 suggest that the nominal model with order $n = 5$ is the most accurate isolator model constructed through this effort. Additional support for this choice is provided by considering together the singular values and frequency
responses of the nominal models in the candidate model set. As noted when discussing the frequency response of the uncertain models (Section 3.3.3), large variations are observed in the frequency response when the model order increases. This is related to an observation made about the HSVs of these larger-order models: a sharp decrease in the magnitude of the HSVs is observed between the fifth and sixth HSV, with higher-order HSVs maintaining similar magnitudes, as can be seen in Figures 3.7 and 3.8 in Section 3.1.2. When reducing a model utilizing balanced truncation, the HSVs are used to identify the states in a balanced realization that may be truncated without losing significant accuracy. If the higher-order states were necessary to capture the dynamics of the system, it would be expected that the HSVs would continue to decrease in magnitude. Since they do not, this suggests that only five states are necessary for capturing the system dynamics and that additional states represent unneeded complexity. Comparison of the simulation results for these models also shows the presence of small oscillations for the $n = 30$ model that are not present for the $n = 5$ model. These are non-physical, and undesirable, artifacts from the additional, unneeded states in the larger model.

Selection of the order $n = 5$ nominal model helps avoid the unnecessary complexity of a larger order model, including the need for model reduction to obtain a COM. In addition, the uncertainty is concentrated in the expected frequency range, so that the sampled models cover the uncertainty well. A disturbance rejection controller will be designed following an $H_{\infty}$ design methodology that acts to anchor the shock at $z = z_{sh,\text{offset}}$ in the presence if input perturbations. After validation on the nominal model, the closed-loop system will be analyzed for robustness against model uncertainty, where the uncertainty has the unstructured representation developed in this chapter.
Chapter 4: Control Design for Disturbance Rejection

The candidate model developed in Chapter 3 using system identification techniques represents a model of perturbations in the shock wave location, \( \tilde{z}_{sh} \), due to input perturbations, \( \tilde{p}_b \). The model does not capture the dynamics of the unstarted system, so a controller is necessary to prevent the shock from propagating far enough upstream that it dislodges from the isolator. The simplest control that can be designed is one that minimizes a given norm of the effects of input perturbations on the shock location, such as a disturbance rejection controller using an \( H_\infty \) technique like those described in Refs. [46, 47, 51–57]. In particular, the two-Riccati equation solution presented by Glover and Doyle and Doyle, et al. is used here to construct a controller that acts to anchor the shock wave at \( z = z_{sh,\text{offset}} \) [53, 55].

In order to satisfy the rank requirements in the \( D \) matrix of the four-block problem (required for the control design algorithm to be implemented), the open-loop system requires modification through the inclusion of filters on various input and output signals. After describing these changes, the open-loop and closed-loop formulation of the system will be developed in order to design the \( H_\infty \) controller utilizing the RCT in MATLAB®. After designing the controller, its ability to attenuate disturbances can be verified through through simulation of the (nominal) closed-loop model with various constant input disturbances and the robust stability analyzed by functions in the RCT and through simulation of sampled models. Validation continues through
implementing the controller in closed-loop with the one-dimensional Euler-equation-based PBM described in Chapter 2. These simulations suggest that the controller is effective in rejecting disturbances, but also reveal some of the limitations of the PBM due to inaccuracy in the model.

4.1 Problem formulation

The closed-loop model schematic in Figure 3.1 required modification in order to develop the four-block problem formulation:

\[ G = \begin{bmatrix} A_G & B_{G,1} & B_{G,2} \\ C_{G,1} & D_{G,11} & D_{G,12} \\ C_{G,2} & D_{G,21} & D_{G,22} \end{bmatrix} \]  

(4.1)

from which the partitions of \( B_G, C_G, \) and \( D_G \) necessary for solving the \( H_\infty \) design problem can be identified. The two-Riccati equation design algorithm assumes that these partitions meet several requirements: the controllability and observability of \((A_G, B_{G,1}, C_{G,1})\), the stabilizability and detectibility of \((A_G, B_{G,2}, C_{G,2})\), and rank requirements on \( D_{G,12} \) and \( D_{G,21} \) \([46, 47]\). The requirements on the triplets \((A_G, B_{G,1}, C_{G,1})\) and \((A_G, B_{G,2}, C_{G,2})\) are met because the nominal plant is controllable and observable, but in order to meet the rank requirements on \( D_{G,12} \) and \( D_{G,21} \) it is necessary to re-define the problem through the inclusion of the filters \( W_1, W_2, \) and \( W_3 \) as shown in Figure 4.1. The filter on the output, \( W_3 \), does not appreciably affect the structure of the partitions in equation (4.1), but filters \( W_1 \) and \( W_2 \) (on sensor noise and control input) introduce non-zero elements to \( D_{G,12} \) and \( D_{G,21} \) to avoid singularities in the formulation that can lead to problem during control design. Note that this figure indicates that the input perturbation, \( \tilde{p}_b \), is caused by pressure
changes in the combustor; the combustor is not modeled in this effort, but this figure suggests how a combustor model may be cascaded with the closed-loop isolator system to form a complete model of the scramjet engine.

### 4.1.1 Open-loop model formulation

For simplicity in developing a state-space formulation of the open-loop system, $W_1$, $W_2$, and $W_3$ are considered as memoryless feedthrough systems (that is, $A_{1,2,3} = B_{1,2,3} = C_{1,2,3} = 0$ and $D_1 = \eta$, $D_2 = \epsilon$, and $D_3 = \xi$). In particular, $W_3$ is assumed to be a direct feedthrough ($\eta = 1$); the inputs to the system are $\bar{p}_b$, $n$, and $u$ (input disturbance, sensor noise, and control input) and the outputs are $z_2$, $z_3$, and $y_1 = \bar{z}_{sh} + \xi n$ (filtered control input, filtered output, and controller input/measured output). The plant has dynamics described by the LTI model in equation (3.1), repeated here.

Figure 4.1: Block diagram of the (nominal) closed-loop DBM, where $P$ is the plant, $K$ is the controller, and the filters $W_1$, $W_2$, and $W_3$ are included to ensure rank requirements on $D_P$ are met.
(recall that $D = 0$ in the isolator model):

\[ \dot{x} = Ax + Bu_p \]

\[ = Ax + B(\tilde{p}_b - u) \]

\[ y = Cx \]

and the outputs are given by the following:

\[ z_2 = W_2 u = \epsilon u \]

\[ z_3 = W_3 y = y \]

\[ y_1 = y + W_1 n = Cx + \xi n \] \hspace{1cm} (4.2)

Defining the input vector, $[\tilde{p}_b \ n \ u]^T$, and output vector, $[z_2 \ z_3 \ y_1]^T$ allows for the open-loop system to be represented by

\[ \dot{x} = Ax + \begin{bmatrix} B & 0 \end{bmatrix} \begin{bmatrix} \tilde{p}_b \\ n \\ u \end{bmatrix} \]

\[ \begin{bmatrix} z_2 \\ z_3 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ C \\ C \end{bmatrix} x + \begin{bmatrix} 0 & 0 & \epsilon \\ 0 & 0 & 0 \\ 0 & 0 & \xi \end{bmatrix} \begin{bmatrix} \tilde{p}_b \\ n \\ u \end{bmatrix} \] \hspace{1cm} (4.3)

where the partitions indicated in equation (4.3) define the partitioned matrices of the four-block problem in equation (4.1). Being able to identify each submatrix of the four-block problem allows for the two-Riccati equation algorithm to be applied to this open-loop system to construct a disturbance rejection controller.
4.1.2 Closed-loop model formulation

The dynamics of the model in Figure 4.1 can be found by ‘closing-the-loop’ in equation (4.3) with a controller $K$, which is characterized by the following state-space model:

$$\dot{x}_k = A_k x_k + B_k y_1$$

$$= A_k x_k + B_k (y + \xi n)$$

$$= A_k x_k + B_k C x + B_k \xi n \quad (4.4)$$

$$u = C_k x_k + D_k y_1$$

$$= C_k x_k + D_k C x + D_k \xi n$$

Substituting the expression for $u$ in equation (4.4) into the input vector in equation (4.3), simplifying the expressions for $x$, $z_2$ and $z_3$, and augmenting the state $x_k$ to the system gives the following closed-loop dynamics:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_k \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} A - B D_k C & B C_k \\ B_k C & A_k \end{bmatrix} \begin{bmatrix} x \\ x_k \end{bmatrix} + \begin{bmatrix} B & \xi B D_k \\ 0 & B_k \xi \end{bmatrix} \begin{bmatrix} \tilde{p}_b \\ n \end{bmatrix} + \begin{bmatrix} \epsilon D_k C & \epsilon C_k \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_k \end{bmatrix} + \begin{bmatrix} 0 & \epsilon \xi D_k \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p}_b \\ n \end{bmatrix} \quad (4.5)$$

where the output $y_1$ is not included in this expression because it is the same as $z_3$ when $n = 0$ (as considered here).

In the frequency domain, the closed-loop formulation is more straight-forward and can be found by examining Figure 4.1; of particular interest is the transfer function from input disturbance to plant output, which will be minimized by the design of an $H_\infty$ controller. Although the control design methodology presented here focuses on the state-space formulation of the model, the frequency domain formulation of the
closed-loop model is presented to identify the transfer function minimized during control design, \( T_{\tilde{p}_b, z_3} \). This transfer function can be built up from the transfer functions that describe each block in the figure:

\[
y = P u_p \\
= P(\tilde{p}_b - u) \\
= P\tilde{p}_b - PK(y + W_1 n) \\
= P\tilde{p}_b - PKy - PK\xi n \\
(1 + PK)y = P\tilde{p}_b - PK\xi n \\
y = (1 + PK)^{-1}P\tilde{p}_b - (1 + PK)^{-1}PK\xi n \\
z_3 = W_3 Pu_p \\
= P(\tilde{p}_b - Ky - Kn) \\
= P\tilde{p}_b - PK(1 + PK)^{-1}P\tilde{p}_b - PK(1 + PK)^{-1}PK\xi n \\
= P(1 - K(1 + PK)^{-1}P)\tilde{p}_b - PK(1 + PK)^{-1}PK\xi n
\] (4.6)

From equation (4.6), the transfer function from \( \tilde{p}_b \) to \( z_3 \) can be found: \( T_{\tilde{p}_b, z_3} = P(1 - K(1 + PK)^{-1}P) \).

### 4.1.3 Adding uncertainty to the model

In order to show the ability of the controller to attenuate input disturbances in the presence of model uncertainty, equations (4.3) and (4.5) should be modified to include uncertain elements in the model. As described in Section 3.3, the nominal model was modified to include a representation of uncertainty constructed from the standard deviation of the frequency response data. For the model considered here, the unstructured uncertainty has multiplicative structure \( P_\Delta = P(1 + W_\Delta) \) and the block diagram for this uncertain model can be seen in Figure 4.2. The uncertainty block
contains a filter, $W$, and a uncertain parameter, $\Delta$, where $|\Delta| \leq 1$; the frequency response of the filter can be seen in the upper-left plot of Figure 3.36 where $\Delta$ is an uncertain gain. As can be seen from Figure 3.27, this model has uncertainty concentrated at high frequencies.

The state-space formulation of the open- and closed-loop uncertain models can be constructed as for the nominal model. The dynamics of the filter, $W$, are introduced by augmenting a state $x_\Delta$ to the system

$$\begin{align*}
\dot{x}_\Delta &= A_\Delta x_\Delta + B_\Delta \bar{u}_p \\
&= A_\Delta x_\Delta + B_\Delta (\tilde{p}_b - u) \\
u_\Delta &= C_\Delta x_\Delta + D_\Delta \bar{u}_p \\
&= C_\Delta x_\Delta + D_\Delta (\tilde{p}_b - u)
\end{align*}$$

The input to the isolator model, $\bar{u}_p$, includes contribution from this uncertain model, as well as from the input disturbance and controller: $\bar{u}_p = \tilde{p}_b + u_\Delta + u$. Forming the augmented state matrix $[x^T \ x_\Delta^T]^T$ and considering the inputs $[\tilde{p}_b \ n \ u]^T$ and outputs $[z_2 \ z_3 \ y_1]^T$, the state-space formulation of the open-loop uncertain model can be found

Figure 4.2: Block diagram of the closed-loop DBM, with unstructured multiplicative uncertainty, where the filters $W_1$, $W_2$, and $W_3$ are included to ensure rank requirements on $D_p$ are met.
as follows

\[
\begin{bmatrix}
\dot{x} \\
\dot{x}_\Delta \\
\dot{x}_k
\end{bmatrix} =
\begin{bmatrix}
A & BC_\Delta \\
0 & A_\Delta \\
B_k C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
x_\Delta \\
x_k
\end{bmatrix} +
\begin{bmatrix}
B(1 + D_\Delta) & 0 \\
-B_\Delta & 0 \\
0 & -B_\Delta
\end{bmatrix}
\begin{bmatrix}
\tilde{p}_b \\
\tilde{p}_b \\
n/u
\end{bmatrix}
\]

(4.8)

\[
\begin{bmatrix}
z_2 \\
z_3 \\
y_1
\end{bmatrix} =
\begin{bmatrix}
0 & 0 \\
C & 0 \\
C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
x_\Delta \\
x_k
\end{bmatrix} +
\begin{bmatrix}
0 & 0 & \epsilon \\
0 & 0 & 0 \\
0 & \xi & 0
\end{bmatrix}
\begin{bmatrix}
\tilde{p}_b \\
\tilde{p}_b \\
n/u
\end{bmatrix}
\]

(4.9)

from which the four-block problem might be redefined to include model uncertainty.

To evaluate the robust stability and performance of the uncertain model in equation (4.8), the controller constructed for the nominal model (equation (4.4)) is implemented in the uncertain model. The dynamics of this controller are augmented to equation (4.8), so that the dynamics of the closed-loop uncertain model can be written as

\[
\begin{bmatrix}
\dot{x} \\
\dot{x}_\Delta \\
\dot{x}_k
\end{bmatrix} =
\begin{bmatrix}
A - B(1 + D_\Delta)D_k C & BC_\Delta & \\
-B_\Delta D_k C & A_\Delta & -B_\Delta C_k \\
B_k C & 0 & A_k
\end{bmatrix}
\begin{bmatrix}
x \\
x_\Delta \\
x_k
\end{bmatrix} +
\begin{bmatrix}
B(1 + D_\Delta) & 0 & \\
-B(1 + D_\Delta)D_k \xi & -B_\Delta D_k \xi & \\
0 & B_k \xi
\end{bmatrix}
\begin{bmatrix}
\tilde{p}_b \\
\tilde{p}_b \\
n
\end{bmatrix}
\]

(4.9)

\[
\begin{bmatrix}
z_2 \\
z_3 \\
y_1
\end{bmatrix} =
\begin{bmatrix}
\epsilon D_k C & 0 & \epsilon C_k C & 0 & 0 \\
0 & \end{bmatrix}
\begin{bmatrix}
x \\
x_\Delta \\
x_k
\end{bmatrix} +
\begin{bmatrix}
0 & \epsilon \xi D_k & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\tilde{p}_b \\
n
\end{bmatrix}
\]

Uncertainty enters the closed-loop model through the system matrices of the uncertain model; in this model, $B_\Delta$ and $D_\Delta$ contain the uncertainty, so the state and input matrices of the closed-loop model contain the uncertainty. This means that the dynamics are directly affected by the model uncertainty, while the output equations are only affected through changes in the state vector related to the modified
dynamics, so that only the state and input matrices differ between models sampled from the uncertain model. Because uncertainty is concentrated at high-frequencies, it is expected that the sampled models produce similar results in steady-state, but different transient responses. This will be shown as the case in Section 4.2.2, where results from simulation of the nominal model are compared to those from simulation of several sampled models.

4.1.4 Design goal

The general problem statement for (suboptimal) $H_\infty$ control design is to find a controller $K$ that minimizes the $H_\infty$-norm of $T_{\tilde{p}_b,z_3}$ in closed loop; specifically, to find a controller such that $||T_{\tilde{p}_b,z_3}||_\infty < \gamma$ for the lowest attainable $\gamma > 0$ [46, 47]. Since $T_{\tilde{p}_b,z_3}$ describes the influence that the input disturbance has on the perturbation of the shock wave from a steady-state condition, a controller that minimizes the norm of this transfer function rejects this disturbance so that the shock wave is essentially anchored at the steady-state location $z = z_{sh,offset}$.

Several algorithms have been developed and studied for solving this control problem, including the original model-matching problem described by Francis [51], the use of linear matrix inequalities proposed by Iwasaki [56], and the celebrated algorithm presented by Glover and Doyle, Doyle, et al., Zhou, et al., and Zhou and Doyle that uses the solution of two Riccati equations to construct the controller [46, 47, 53, 55]. The model-matching problem is difficult to implement for a strictly-proper model ($D = 0$), such as the isolator model constructed in Chapter 3, as it requires extracting the inner-outer factorization of a strictly-proper system; although methodologies for computing this factorization are presented in Refs. [58–62], these require either a proper, but not strictly-proper, model or a model with rank restrictions on other
matrices. In addition, this design methodology utilizes few built-in MATLAB® functions, leaving room for error in the implementation of the factorization algorithms.

The RCT in MATLAB® contains the function *hinfsyn*, which performs $H_\infty$ design through implementation of the two-Riccati equation design technique in Refs [46, 47, 53, 55]. Requirements on the four-block formulation (equation (4.1)) are mentioned in Section 4.1; assuming these conditions hold, the disturbance rejection controller can be found following iterative techniques to minimize $\gamma$, the upper bound on $\|T_{\bar{p}_b,z_3}\|_\infty$.

Two $\gamma$-dependent Hamiltonians, $H$ and $J$, are constructed as follows

$$
H = \begin{bmatrix}
    A & \gamma^{-2}B_1B_1^T - B_2B_2^T \\
    -C_1^TC_1 & -A^T
\end{bmatrix}
$$

$$
J = \begin{bmatrix}
    A^T & \gamma^{-2}C_1^TC_1 - C_2^TC_2 \\
    -B_1B_1^T & -A
\end{bmatrix}
$$

which represent Riccati equations whose positive definite solutions are denoted by $X_\infty$ and $Y_\infty$, respectively. For a given $\gamma > 0$, these solutions are checked for positive definiteness ($X_\infty > 0$, $Y_\infty > 0$), and the spectral radius of the product is verified to be less than $\gamma^2$, that is, $\rho(X_\infty Y_\infty) < \gamma^2$ [46, 47]. The suboptimal $H_\infty$ controller is found through iteration on $\gamma$; specifically, *hinfsyn* uses bisection to minimize $\gamma$ (within a specified tolerance) [49, 50]. Because this algorithm is included as part of the RCT, the implementation has been well-studied and is guaranteed to produce an $H_\infty$ controller that minimizes the influence of input disturbances on the output when the assumptions on the controllability, observability, stabilizability, detectibility, and rank on various partitions of the four-block problem hold.
4.2 Control design and validation

The (nominal) isolator model for which a controller is to be designed has a state with dimension $x \in \mathbb{R}^n$, where $n = 5$; it has been shown that this model is able to capture the motion of a shock, as it propagates upstream, to within 2 cm. Because the data used to construct the model was obtained through CFD simulation with inviscid flows, it was not possible to include cases with downstream shock propagation (a decrease in backpressure), so the dynamics of the isolator model due to decreases in backpressure could not be validated; however, two such cases were examined in validating the ability of the controller to attenuate input disturbances.

To design the controller, characterization of the four-block problem in equation (4.3) was completed by choosing $\epsilon > 0$ and $\xi > 0$; this controller has the same state dimension as the model (that is, $n_k = 5$). Specifying a $\gamma$ tolerance of $10^{-10}$ to ensure an accurate estimation of $\gamma$, hinfsyn was used to solve for $K$ and the value of $\gamma$ that results from the iteration. A series of controllers was constructed by varying $\epsilon$ and $\xi$ in order to see how the choice of scaling on the feedthrough affects $\gamma$ and the stability of the controller; a summary of the results is in Table 4.1. It can be seen that the choice of $\epsilon$ and $\xi$ affects the stability of both the controller and the closed-loop system, in addition to affecting $\gamma$. When $\xi$ is fixed (at 0), decreasing $\epsilon$ causes $K$ and the closed-loop system to have fewer unstable modes (increasing $\epsilon$ causes instability in closed-loop). For this case, the best results were obtained when $\epsilon = 10^{-6}$ ($\gamma = 0.0257$); decreasing $\epsilon$ further did not significantly improve $\gamma$ and led to problems in the design algorithm. Fixing $\epsilon$ at $10^{-6}$, the effect of varying $\xi$ was studied; it was found that increasing $\xi$ caused $K$ to have more stable modes and that $\xi = 4 \cdot 10^{-2}$ was the smallest value for which $K$ was stable. In this case, $\gamma = 0.0394$, which is larger than when $\xi = 0$, but the stable controller improved the robust stability of the closed-loop system. It should be noted that the cases with $\epsilon = 10^{-6}$ and $\xi = 0$, $\xi = 10^{-6}$,
Table 4.1: Stability results and $\gamma$ for a series of controllers constructed with varying $\epsilon$ and $\xi$. Note that the numbers in parentheses indicates the number of unstable modes.

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>$10^{-6}$</th>
<th>$10^{-4}$</th>
<th>$10^{-2}$</th>
<th>$10^{-6}$</th>
<th>$10^{-6}$</th>
<th>$10^{-6}$</th>
<th>$10^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$10^{-6}$</td>
<td>$10^{-4}$</td>
<td>$10^{-2}$</td>
<td>$4 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.0257</td>
<td>0.0268</td>
<td>0.026</td>
<td>0.0257</td>
<td>0.026</td>
<td>0.0394</td>
<td>0.0863</td>
</tr>
<tr>
<td>CL stable?</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

and $\xi = 10^{-4}$) produced $\gamma$ which were the same out to the $10^{-6}$-place, and that the two cases with $\xi \neq 0$ were the same out to the $10^{-7}$ place; these differences would not be noticed if a larger tolerance was specified for the control design algorithm.

### 4.2.1 Design and nominal stability

Two controllers were constructed for the nominal model: one with $\xi = 0$ (no sensor noise) and one with $\xi \neq 0$; the first case produces an unstable controller, while the second produces a stable controller and improved robustness of the closed-loop model. Although several values for $\epsilon$ and $\xi$ were considered in constructing Table 4.1, only the two pairs that resulted in stable and unstable controllers with the smallest $\gamma$ were chosen for further analysis. The two-Riccati equation design technique was implemented utilizing the $\text{hinfsyn}$ function in the RCT, with $\epsilon = 10^{-6}$ and $\xi = 0$ for the unstable controller and $\xi = 4 \cdot 10^{-2}$ for the stable controller; these choices produced controllers with $\gamma = 0.0257$ and $\gamma = 0.0394$, respectively. The $H_{\infty}$ controller has the same dimension as the nominal system ($n_k = 5$), so the closed-loop system in equation (4.5) has dimension $n_{\text{CL}} = 10$. In order to evaluate the effectiveness of controller, the closed-loop system is simulated with initial conditions $x = 0$ (that is, $\bar{p}_{b,0} = 0$ and $\bar{z}_{sh,0} = 0$) and the four constant input disturbances in Table 4.2, representing step-inputs that produce shock propagation both upstream
Table 4.2: Step-inputs for validation of the $H_{\infty}$ controller in closed-loop with the nominal isolator model. The disturbances $\tilde{p}_b$ cause the shock to propagate a distance $\tilde{z}_{sh}$ from the steady-state location $z_{sh, offset}$, and it is expected that the controller maintains the shock near $z = z_{sh, offset}$ in the presence of the input disturbances.

<table>
<thead>
<tr>
<th>$p_b$ (psi)</th>
<th>$\tilde{p}_b$ (psi)</th>
<th>$z_{sh}$ (m)</th>
<th>$\tilde{z}_{sh}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.702</td>
<td>-1.298</td>
<td>0.85065</td>
<td>0.20995</td>
</tr>
<tr>
<td>32.824</td>
<td>-0.176</td>
<td>0.66369</td>
<td>0.02299</td>
</tr>
<tr>
<td>34.197</td>
<td>1.197</td>
<td>0.42774</td>
<td>-0.21296</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>0.31033</td>
<td>-0.33037</td>
</tr>
</tbody>
</table>

and downstream. (The perturbations $\tilde{p}_b$ and $\tilde{z}_{sh}$ are measured from the offsets used to construct the DBM, which are in Table 3.1.) Note that all simulations presented here were run with $n = 0$ (no sensor noise).

The results in Figure 4.3 demonstrate the ability of the controller to maintain the steady-state shock location at $z = z_{sh, offset}$; the subplots in the figure correspond to $\tilde{p}_b = -1.298$ psi, $-0.176$ psi, 1.197 psi, and 2 psi, respectively. It can be seen that a 2 psi input disturbance results in a steady-state error near 3 cm, which is slightly greater than the error found in validation of the nominal model (Section 3.2) and only 26.1% of the average duct-height, $D_H = 11.5$ cm. This is less than 10% of the distance which the shock propagates in the uncontrolled model, where the steady-state shock location is around 13 cm upstream of the inlet. In particular, for all four simulations, the shock propagated in the closed-loop model with the unstable controller was 9.35% of the distance expected from the uncontrolled model and 8.76% of this distance with the stable controller.

It is also interesting to note how the direction of shock propagation is affected by the stability of the controller. The unstable controller overcompensates for the input disturbances, causing shock propagation in the opposite direction as expected: upstream for negative perturbations and downstream for positive perturbations. In
Figure 4.3: Perturbations of shock waves, measured from $z_{sh,\text{offset}}$, for simulation of the closed-loop nominal model with both stable and unstable controllers. The input perturbations for each simulation are $\tilde{p}_b = -1.298$ psi, $-0.176$ psi, $1.197$ psi, and $2$ psi, respectively.

In contrast, the stable controller does not quite attenuate the disturbance, as the shock propagates a small distance downstream (upstream) for a negative (positive) input disturbance. Figure 4.3 also shows that, for the first few milliseconds of each simulation, shock propagation is nearly the same for both controllers, producing a sudden change in the direction of propagation in establishment of a steady-state location in simulation with the unstable controller. For both controllers, however, it takes around 15 ms for a steady-state shock location to be established.

The disturbance rejection characteristics of the controller can be more easily viewed in Figure 4.4, where shock locations are calculated by adding $z_{sh,\text{offset}}$ to the model output. These plots are arranged similar to Figure 4.3, where the input disturbances are $\tilde{p}_b = -1.298$ psi, $-0.176$ psi, $1.197$ psi, and $2$ psi, respectively. Included
Figure 4.4: Shock wave locations for simulation of the uncontrolled and controlled nominal models, with both stable and unstable controllers. The dotted line indicates the initial location of the shock wave. The input perturbations for each simulation are $\tilde{p}_b = -1.298$ psi, $-0.176$ psi, $1.197$ psi, and $2$ psi, respectively.

with results for the closed-loop isolator model are results for simulation of the uncontrolled model with each input, which reinforces conclusions drawn from Figure 4.3. Although the disturbances are not perfectly attenuated, the shock wave is maintained within a small region about $z = z_{sh, offset}$, even when the input disturbance would otherwise cause the shock wave to propagate 30 cm from this location. The range of input perturbations for which the isolator would remain started is only slightly larger than the range of $\tilde{p}_b$ for which the model was simulated, suggesting that both the stable and unstable controllers are able to reject these disturbances, keeping the shock wave from propagating more than 3 - 4 cm from the steady-state location, well within one duct height.

In addition to the simulation results, it is also of interest to examine the control effort necessary to achieve the disturbance rejection results in Figures 4.3 and 4.4.
Figure 4.5 shows the input disturbance and the corresponding control efforts necessary for each of the four simulations (that is, \( \tilde{p}_b = -1.298 \) psi, \(-0.176 \) psi, \(1.197 \) psi, and \(2 \) psi, respectively). It can be seen that the control effort is on the same order as the input disturbance and that the magnitude of the effort necessary from the stable controller is always less than \( \tilde{p}_b \), while that of the unstable controller is always greater than \( \tilde{p}_b \). This accounts for the fact that the unstable controller produces a shock wave that propagates in the opposite direction as expected, since the input to the model is \( \tilde{p}_b - u \). In general, the stable controller requires less effort (smaller input signal) than the unstable controller, a characteristic most noticeable in the transient dynamics of the input. The unstable controller shows large variation in the first 15 ms of simulation, reaching peaks several times larger than the steady-state value as it works to reject the input disturbance. In contrast, the stable controller does not reach comparable large peaks during the same time frame, so that less control effort, overall, is necessary for this controller to attenuate the disturbance.

### 4.2.2 Robustness analysis

After verifying that the controllers designed in Section 4.2.1 minimized the affects of input disturbances on the output for the nominal model, it was necessary to study how robust the closed-loop model is to uncertainty and variation in the dynamics. The closed-loop model in equation (4.9) was constructed for both the stable and unstable controllers; it should be noted that the formulation of the uncertainty in these models was changed from that discussed in Section 3.3. Instead of allowing \( \Delta \) to be an unknown complex gain, with \(|\Delta| \leq 1\), \( \Delta \) was restricted to be an unknown real gain; this did not significantly affect the frequency responses of the uncertain (open-loop) model, as can be seen in Figure 4.6, where the expected upper and lower bounds on the magnitude are indicated as red and cyan lines, respectively. In closed-loop, the
gain bounds on $\Delta$ were affected by the controller: when $K$ was unstable, uncertainty was limited to $|\Delta| \leq 0.35$ in order for the closed-loop model to be guaranteed stable (stable closed-loop models could be found for $|\Delta| > 0.35$, but it was also possible for a model to be unstable in this range), while a stable $K$ allowed for stable closed-loop models to result for all gains on $|\Delta| \leq 1$.

Having established bounds on $\Delta$, the RCT was used to analyze stability limitations on the closed-loop model. A function is available that produces workspace output describing the robust stability of the uncertain model, however it was found that the results from analysis of the model with the unstable controller did not agree with what was found through simulation. Specifically, the uncertain model was claimed to be stable for any $\Delta$, but some models became unstable when $|\Delta| > 0.35$, as previously discussed. In contrast, the same analysis suggests the closed-loop model with the stable controller will be stable for any $\Delta$ under $\Delta = 3.6347$. The model was also

Figure 4.5: Input disturbances (solid line) and control efforts (dashed lines) for simulation of the closed-loop nominal model with both stable and unstable controller.
Figure 4.6: Bode plots of the modified uncertain (open-loop) model, where the complex gain has been replaced with a real gain. The expected upper and lower bounds (one standard deviation of the magnitude response) are shown as red and cyan lines, respectively, and it can be seen that this modification does not affect the ability of the model to cover the uncertainty.

analyzed to find the value of $\Delta$ that produces the worst-case gain in the model. For both models, this occurred for a $\Delta$ on the edge of the range: $\Delta = 0.35$ for the model with unstable controller and $\Delta = -1$ for the model with stable controller. This $\Delta$ was then used to construct the closed-loop model with the worst-case gain characteristics (on the defined range of $\Delta$) (the ‘worst-case’ model; this model, along with a set of model sampled from the uncertain model, was simulated with the four inputs in Table 4.2.

    Shock perturbations for the worst-case model are shown in Figure 4.7 with nominal simulation results. It can be seen from these plots that uncertainty in the model does not affect the steady-state shock perturbation, but does have effect on the transient behavior. This is expected, since uncertainty is concentrated in high-frequency ranges, as can be seen in Figure 4.6, and the transient behavior is characterized by
Figure 4.7: Shock perturbations for simulations of the closed-loop uncertain models with stable and unstable $H_\infty$ controllers. Simulation results of the nominal model and the model with worst-case gain are present. The input perturbations for each simulation are $\tilde{p}_b = -1.298 \text{ psi}, -0.176 \text{ psi}, 1.197 \text{ psi}, \text{ and } 2 \text{ psi, respectively.}$

high-frequency oscillations. Figure 4.8 offers a closer look at the first 10 ms of the simulation, where the most prominent transient behavior can be observed. In this figure, simulation results are also shown for five models sampled from the uncertain model. The shock perturbations are generally bounded by the nominal and worst-case simulation results for both the stable and unstable controllers. Because of the sampling time of the simulation (the isolator models are in the DT domain), these closer looks suffer from some inaccuracy due to the unrefined time-step. Even so, it is possible to conclude that, for the given ranges of $\Delta$, these models are robust to changes in the model (in the presence of constant input disturbances).

As with the nominal model, it is of interest to look at the control effort necessary to achieve the level of disturbance attenuation observed in the closed-loop models. Figure 4.9 shows the control inputs for the nominal and worst-case models with the
unstable and stable controllers. Here, it can be seen that the worst-case models are characterized by transient oscillations with larger magnitude than those present in the nominal control inputs. The transient in the control input for the model with the unstable controller takes much longer to dissipate than the transients in any of the other cases shown in the figure; this may be due to the fact that this model is closer to being unstable than the other models, suggested by the smaller range of $\Delta$ possible before the model goes unstable. Like the simulation results, these plots show there is not much difference in steady-state between the nominal and worst-case models, as expected due to the characterization of uncertainty.

Figures 4.10 and 4.11 provide a closer look at the transient behavior of the control efforts for the nominal, worst-case, and five sampled models. As observed in Figure 4.8, the control efforts for the five sampled models are (generally) bounded by
Figure 4.9: Input disturbances (dotted line) and control efforts (solid lines) for simulation of the closed-loop uncertain models with stable and unstable $H_\infty$ controllers. Control efforts are shown for simulation of the nominal closed-loop models and the models with the worst-case gain.

The control efforts necessary for the nominal and worst case models. In addition, the maximum control effort necessary for the model with the unstable controller is over twice that necessary for disturbance rejection in the model with a stable controller. This, along with the improved range of $\Delta$ for which the closed-loop model remains stable, suggests that the stable controller may be the most useful, despite having a slightly larger $\gamma$.

4.2.3 Validation on the reduced-order physics-based model

The stable controller constructed for the nominal DBM was next implemented in closed-loop with the PBM constructed from the one-dimensional Euler equations (Section 2.3); the block diagram in Figure 4.12 shows a schematic of this system. Because the input of the PBM is backpressure and the output is an array of pressure
Figure 4.10: A closer look at the first 10 ms of inputs in Figure 4.9 for the uncertain, closed-loop model with an unstable controller. Also shown are the results for simulation of five sampled models.

Figure 4.11: A closer look at the first 10 ms of inputs in Figure 4.9 for the uncertain, closed-loop model with a stable controller. Also shown are the results for simulation of five sampled models.
measurements (a pressure profile), it is necessary to include several additions to obtain the block diagram of the closed-loop system. The most important modification is a routine that extracts the shock wave location from the pressure profile (the ‘getShLoc’ block), which uses the same methodology to determine the shock location as used to construct the input/output data in Chapter 3: construct the pressure ratio profile and determine the location at which it exceeds 1.1 (pressure has increased 10% over the tare value). Since the DBM captures the relationship between input perturbations and output perturbations, it is necessary to remove $z_{sh,offset}$ from the estimated shock location, and add $p_{b,offset}$ to the input signal, for the controller to be implemented correctly. The final modification is a gain to convert the input to the PBM from psi to Pascals in order to match the units for which the initial conditions of the PBM are defined. It should also be noted that the controller was converted from DT to CT, using a bilinear transformation, because the PBM is a CT model.

Simulation of this closed-loop model revealed several limitations in the PBM. Most notable was the inability to validate the model in the presence of a decrease in backpressure; when the scaling functions developed in Section 2.3.2 were implemented, the simulation encountered numerical errors as soon as a decrease in backpressure was
detected. This was confirmed when attempting to run simulations with a decrease in backpressure and $\alpha_{sc}$ fixed to a value on the order of $10^5$, where these same errors were encountered. This further limits the accuracy of the PBM, as it is required that $\alpha_{sc}$ and $\beta_{sc}$ be fixed, which has been shown to affect the steady-state shock location for a given backpressure, or that only $\alpha_+^{sc}(p_{in})$ be implemented. In addition, validation was limited to cases of increasing backpressure, similar to the restriction in place when running CFD simulations with inviscid flow.

Two simulations were run with fixed $\alpha_{sc}$ and $\beta_{sc}$, one with $\tilde{p}_b = 1.197$ psi, $\alpha_{sc} = 310$, and $\beta_{sc} = 50,000$, and the other with $\tilde{p}_b = 2$ psi, $\alpha_{sc} = 290$, and $\beta_{sc} = 50,000$. The fixed values of $\alpha_{sc}$ were chosen as those producing the most accurate steady-state shock locations for the given input disturbance in simulation of the uncontrolled PBM. Although $\beta_{sc} = 0$ for the simulations discussed in Section 2.3.2, a non-zero value was assigned for the closed-loop simulations in order to avoid potential numerical errors. Two additional simulations were run for these same input disturbances with the second-order damping matrices scaled by $\alpha_+^{sc}(p_{in})$ and coefficients from fit 3 in Table 2.9. Simulations were attempted with the coefficients from fit 2, but after 20 - 25 ms of data was collected, numerical problems were encountered and the control input showed oscillations with increasing amplitudes, suggesting this fit would not be a good choice. Results of these simulations are shown in Figure 4.13, where the results from simulation of the (nominal) DBM are shown for comparison. The expected loss of accuracy in the PBM can immediately be observed from the figure, where the steady-state shock locations from simulations of the PBM are approximately 4 cm upstream of those predicted by the DBM, but still less than the one duct height the shock wave is expected to oscillate through in steady-state ($0.608 \bar{D}_H$). The oscillations observed during these simulations were due to the algorithm for calculating the shock location rather than due to actual oscillation of the shock wave, which are limited
Figure 4.13: Location of the shock wave from simulation of the closed-loop PBM with the stable $H_\infty$ controller. The dashed lines show the results of the same simulations on the DBM and are included for comparison. Note that only $\tilde{p}_b > 0$ are able to be simulated for the closed-loop PBM.

by the spatial resolution of the isolator grid (Figure 2.2). Implementing the PBM with an input-dependent scaling function did not significantly affect the accuracy of the closed-loop model; in fact, the steady-state shock location was the same for both simulations (about half-way between the steady-state locations from simulations with fixed $\alpha_{sc}$). These simulations showed overshoot of the steady-state location, about 1.5 cm ($0.13\bar{D}_H$) for the simulation with $\tilde{p}_b = 1.197$ psi and 4 cm ($0.347\bar{D}_H$) for simulation with $\tilde{p}_b = 2$ psi, with the shock propagating upstream for the first 7 ms of simulation before propagating downstream to establish a steady-state 9 - 10 ms into the simulation. This may be due to the relation between $\alpha_{sc}$, the backpressure input to the PBM, and the shock propagation in the isolator.

Since the steady-state shock location is further upstream than that predicted from simulation of the DBM, it is expected that the control effort necessary to achieve
these results is larger than for the DBM; Figure 4.14 shows that this is the case. In general, the control effort for simulation of the closed-loop PBM is more than twice that necessary for simulation of the closed-loop DBM, but under 5 psi for both input disturbances under consideration. One drawback of these control signals is the oscillations in steady-state; these may be due to the conversion of the controller to CT and it appears they may dampen out if the simulation were run for a long enough time.

For simulation with the input-dependent scaling function, the control efforts reached a peak around 7 ms into the simulation, before oscillating about the same steady-state value 3 ms later. This is reflected in the propagation of the shock wave during the simulation (Figure 4.13), where the steady-state shock location for both simulations was the same. While the control effort shows oscillation of the same frequency, the amplitude of oscillation is larger for $\tilde{p}_b = 2$ psi, which is expected because it is the larger input disturbance. But like the other simulations of the closed-loop PBM, the amplitude of the oscillations is slowing decreasing, suggesting that these oscillations would eventually dampen out if the simulation were run for a long enough time.

The effect of changing the initial conditions of the simulation was studied through simulation of the closed-loop PBM with initial conditions corresponding to a back-pressure $p_{b0} = 32.824$ psi, a decrease of 0.176 psi from $p_{b,\text{offset}}$. Simulations were attempted with initial conditions further perturbed from $p_{b,\text{offset}}$, but numerical errors were encountered that terminated the simulation, again highlighting limitations of the PBM. The results in Figure 4.15 are for a input disturbance $\tilde{p}_b = 1.197$ psi (measured from $p_{b,\text{offset}}$), where the ability of the controller to attenuate this disturbance can be observed. As in Figures 4.13 and 4.14, the shock wave overshoots by around 4 cm the steady-state location that it established 9 ms into the simulation. This location is around 5 cm upstream of $z_{sh,\text{offset}}$, which is nearly the same as for simulation with the initial conditions defined by $p_{b0} = p_{b,\text{offset}}$. The control signal similarly shows
Figure 4.14: Control efforts for simulation of the closed-loop PBM with the stable $H_\infty$ controller. The dashed lines show the control efforts for simulation of the DBM for the same inputs and are included for comparison. Note that only $\tilde{p}_b > 0$ are able to be simulated for the closed-loop PBM.

overshoot around 7 ms into the simulation before oscillating about a steady-state location, as was found to be characteristic of the control efforts in simulation with the PBM.

These control design results are encouraging as they suggest the possibility of designing a controller for a model with reduced accuracy (here, the DBM) and implemented it on a higher-fidelity model with small loss in accuracy (here, the PBM). The control problem defined in this chapter is simplified to consider a disturbance rejection controller, with no tracking requirement, the $H_\infty$ controller is shown to be effective in rejecting constant input disturbances for the nominal model, and the closed-loop model is robust against model uncertainty when $K$ is stable. Similar success was found in implementation of the stable controller with the nonlinear PBM, although
the limitations of this model, due to the sensitivity of the scaling factor to the back-pressure and to numerical errors encountered when backpressure was decreased, were magnified. Despite these limitations, the stable $H_\infty$ controller was shown to limit the shock propagation to under 8 cm ($0.695 \bar{D}_H$) in the presence of relatively large input disturbances (when $\tilde{p}_b = 2$ psi, the shock propagates upstream within 13 cm of the isolator entrance plane), meeting the simplified design goal.
Chapter 5: Conclusions and Outlook

Recently renewed interest in scramjet propulsion has revealed the need for development of a control-oriented model of the engine that accurately captures the transient dynamics relevant to the occurrence of unstart. The engine becomes unstarted when perturbations in the flow cause the shock train that has established in the isolator to propagate far enough upstream that it dislodges from, and reforms external to, the inlet. The establishment and propagation of the shock train is a highly nonlinear, complex flow phenomenon that many models are unable to capture as they are developed for simplified systems or designed to capture only steady-state dynamics. Because of its importance in indicating the onset of unstart, a model of the isolator should be able to accurately capture the transient dynamics of the shock as it propagates due to a change in the backpressure imposed on the isolator, but also be of a small order for use in control design. One of the simplest possible controllers is one that acts to anchor the shock wave at a specified location in the presence of input (backpressure) perturbations that may occur during high-demand flight maneuvers and is robust against uncertainty in the model.

5.1 Physics-based models

A series of physics-based models of the scramjet isolator have been constructed through discretization of the Euler equations for compressible flow, following and
expanding the methodology presented by Chicatelli and Hartley [28,29]. These models capture more detail than just the propagation of the shock due to changes in the backpressure, as they allow for measurements of various flow variables (specifically pressure) to be made, from which profiles can be constructed to show the behavior of these variables through simulation.

Direct application of the Chicatelli and Hartley methodology led first to the construction of a linearized, locally-accurate, small-perturbation model from the two-dimensional Euler equations. Although linear, the size of the model \( n = 29204 \) states proved to be prohibitive of quick validation as it took nearly ten hours to obtain results for each simulation despite having optimized the code to run on a high-performance workstation and take advantage of multiple processor cores through the Parallel Computing Toolbox in MATLAB®. Analysis of the simulation results further revealed that small perturbations in the input produced perturbations in the flow that were too large to be captured by the model. This was seen through the inability of the model to capture any shock motion and suggested that it would be necessary to instead consider a nonlinear model of the isolator.

Expansion of the discretized two-dimensional Euler equations (which were linearized to form the SPM) yielded a nonlinear isolator model that was affine with respect to input. Shock motion was not originally captured by this model, but the CFD technique of adding artificial viscosity to simulations of inviscid flow was employed through the introduction of second- and fourth-order damping. These terms included tunable parameters that could be adjusted to determine how much influence the damping would have on the model. Because this model required calculating several \( n \times n \) matrices at each time step, where \( n = 29204 \), it was not possible to efficiently simulate the model: allowed to run for just two weeks, less than 1 ms of data was collected from simulation of the model with damping, from which it was
difficult to draw any conclusions as to the accuracy of the model. This limits the usefulness of such a model in applications where tuning of parameters necessitates a model that can be simulated quickly.

A smaller order nonlinear model can be constructed by applying the methodology of Chicatelli and Harley to the one-dimensional Euler equations, which reduces the number of states by a factor of 67 (from $n = 29204$ to $n = 447$). This model has the same affine form as that constructed from the two-dimensional Euler equations, and a related damping scheme is implemented to complete the characterization, but the reduction in order (and revised boundary condition treatment) allowed simulations to be completed (50 ms long) in under 30 minutes for the case of increasing backpressure and under 3 hours for decreasing backpressure. The quick simulation time allowed for study of the effects of tuning the damping parameters, from which it was concluded that an input-dependent scaling factor for the second-order damping terms was necessary to improve the overall accuracy of the model. After tuning the model to produce an accurate steady-state shock location for several backpressure inputs, fourth-order polynomials were constructed to estimate the scaling factors for the separate cases of backpressure increases from the initial condition and backpressure decreases from the initial condition. These fits were validated through simulation of the model and the most accurate fit for each scaling function was chosen to complete the modified characterization of the nonlinear, physics-based model of the isolator.

### 5.1.1 Outlook

Although it has been shown that this nonlinear model of the isolator can be tuned for accurate approximation of the steady-state shock wave location, simulations have suggested that there is room for improving the model. Specifically, the scaling functions should be made more accurate by constructing them from $\alpha_{sc}$ collected
through validation of the model against a larger set of CFD data, or a set of data that provides a better representation of the shock wave at various steady-state locations in the isolator. The simulation results found here suggest that the scaling functions $\alpha_{sc}^{\pm}(p_{in})$ should have no dependence on the initial conditions and a single function for each direction of change in backpressure can be calculated. Further improvement in the accuracy of the fits may be obtained by adjusting the time step and accuracy of the simulations in SIMULINK®, which may improve the tuning of the model and also reduce the frequency at which numerical errors are encountered.

The steady-state spatial accuracy of the model with these new scaling functions could then be evaluated through comparison of the simulation results to steady-state CFD solutions with the same backpressure change, as discussed here. Validation with more complex input functions, such as multi-step or continuous inputs, should follow; this requires addressing the temporal inaccuracy observed in simulation of the one-dimensional Euler equation based model, where it took 20 ms longer to reach steady-state in simulation than suggested through the CFD simulations. The usefulness of this model in control design applications remains somewhat limited due to these inaccuracies, but the modifications introduced here have improved the linearized model introduced by Chicatelli and Hartley in Ref. [28] and provided an important first-step toward construction of an accurate, control-oriented physics-based model of a scramjet isolator, as previous efforts have produced linearized models of shock dynamics in ramjet inlets [10], steady-state flow correlations [12,13], and steady-state models of scramjet dynamics [15–21].

5.2 Data-based model construction and control design

A set of data-based models was constructed using the N4SID system identification technique and three sets of transient CFD data. Two families of linear, SISO models,
defined by the set of data used for identification, were constructed, where the models in each family were identified by the order and the relative degree. For models with relative degree \( nk < 2 \), both a small-order and a large-order model were constructed, where the large-order model required reduction (via balanced truncation) before use for control design. The four models which proved to be most accurate during validation were further modified through the inclusion of uncertainty, which allows for evaluation of the robustness of a control design. Structured uncertainty in models was present as uncertainty on the elements of the state and input matrices, expressed in canonical coordinates, or as uncertainty in the frequency response of each nominal model. Three methodologies were developed to transform this uncertainty to an unstructured representation in a model with multiplicative uncertainty. It was found that this was most successful for the model with order \( n = 5 \), which is of small enough size to remove the need for reduction for control design.

The general goal of the control design is to prevent unstart during high-demand flight maneuvers that increase the backpressure induced on the isolator enough to dislodge the shock train. This goal was simplified by defining the problem in the framework of a disturbance rejection controller, where the two-Riccati equation design technique described by Glover and Doyle, Doyle, et al., Zhou, et al., and Zhou and Doyle is implemented to design an \( H_\infty \) controller for the DBM [46, 47, 53, 55]. After framing the isolator model in the standard four-block problem framework in which this \( H_\infty \) design technique was developed, two controllers were designed, one stable and one unstable. Both controllers yielded a stable closed-loop system, but it was found that the stable controller yielded a closed-loop model with better stability robustness against model uncertainty. In particular, the uncertain parameter \( \Delta \) could vary across the entire range \([-1, 1]\) for this closed-loop model, while in the model constructed with the unstable controller \( \Delta \) could only vary on the range
Simulation of a set of models sampled from the uncertain closed-loop models revealed that both controllers were able to reject the input disturbances, even for the model with worst-case gain characteristics. The stable controller was further validated through implementation with the PBM, where it was shown that the shock propagation was limited by the controller, with an expected reduction of accuracy, for simulations with both fixed and input-dependent $\alpha_{sc}$.

5.2.1 Outlook

Characterization of the engine model can be completed by cascading a combustor model with the isolator model. This model may be an algebraic correlation or a dynamic system that relates actuation to the backpressure that is imposed on the isolator. The control scheme can then be generalized for use with this complete engine model and evaluated to study and evaluate the ability of the designed controller to meet the control objectives, which include robustness against disturbances in the backpressure and against sensor delay and input saturation. The modeling and control efforts discussed here provide a first-step toward the design of a controller for a COM which can then be validated through implementation with a higher-fidelity model. Although the modeling efforts described in Refs. [22, 23, 27] produce DBMs, these are constructed by performing system identification on experimental results, which is different from the DBM constructed in this thesis, which is the first to use transient CFD data for identification of an isolator model. The design goal considered here, a controller that limits shock propagation to a small distance about a given location $z_{sh, offset}$, provides a simple solution to the idea of preventing unstart in the isolator. A more sophisticated controller may allow for tracking a reference signal or allowing the shock to propagate to a certain point before acting to anchor the shock in a confined
region of the isolator, but it is useful to be able to show that a controller can be designed to meet the simple goal of disturbance rejection.

The success of this design, shown through simulation in closed-loop with both the DBM for which it was designed and the PBM that captures the flow dynamics in the isolator, is accompanied by limitations in the PBM highlighted through validation simulations. In particular is the inaccuracy of the fits for the input-dependent scaling functions, which reduces the accuracy of the simulation. In addition to this, the inability of simulating the closed-loop PBM with a $\bar{p}_b < 0$ without encountering numerical errors suggests that the model should be re-evaluated in how it is formulated for downstream shock propagation. This may help improve the fit for $\alpha_{sc}^{-}(p_{in})$, which was shown to produce numerical errors or inaccurate results during validation of the model. Although these limitations show the need for improving the accuracy of the PBM (in order to improve evaluation of the controller), the results show the promise borne by a model-based control design methodology that is effective in preventing unstart in a model of higher-fidelity. Previous efforts toward implementation of a controller that acts to prevent unstart have involved open-loop designs that involve detection of unstart in the isolator and activation of an actuator to affect the flow and reverse the unstart process [11,24–26], but the effort presented here represents the first comprehensive effort in constructing a control-oriented model of a scramjet isolator and designing a model-based feedback controller that is successful in anchoring the shock wave in the presence of input disturbances.
References


Appendix A: Two-dimensional Euler equation-based nonlinear model

The intermediate PBM of the isolator constructed through discretization of the two-dimensional Euler equations has the form in equation (2.29), repeated here for convenience:

\[
\dot{x} = \bar{A}(x)x + \bar{B}(x,p_{in})
\]

The vector \( \bar{B}(x,p_{in}) \) can be expanded through application of Euler’s theorem of homogeneous functions to construct the state and input matrices, \( A(x) \) and \( B(x) \). This is possible because the boundary conditions are homogeneous functions of \( x(u_{i,k}) \) with degree one.

A.1 Constructing the state matrix

The state matrix \( A(x) \) is constructed through modification of \( \bar{A}(x) \) ((2.40)) in the block rows associated with states \( u_{i,1} \) and \( u_{i,N_x} \), which describe flow conditions at the isolator entrance and exit planes. These modifications are from the state-dependent terms of the input vector \( \bar{B}(x,p_{in}) \), which contains the boundary condition functions.

At the isolator entrance, the flow is assumed to be supersonic across the entire plane and the boundary conditions at these grid points are found in the block rows of \( \bar{B}(x,p_{in}) \) associated with \( k = 1 \). Recalling that the boundary condition function
at the entrance plane is $B_{bc}^{in}(u)$, these block rows can be simplified as follows:

$$-\frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} u$$

$$= -\frac{1}{\Delta T} I u \quad (A.1)$$

Since $B_{bc}^{in}(u)$ has no dependence on the input, these terms do not contribute to $B(x)$.

When the streamwise velocity is subsonic at the exit plane, the input-dependent term $L_B(u, p_{in})$ can be expanded through application of Euler’s theorem of homogeneous functions:

$$L_B(u, p_{in}) = \left[ \begin{array}{c} \left( K_h^{-1} \right)_N \\ \frac{\partial B_{bc}^{out}}{\partial u} \\ -\frac{1}{\Delta T} \frac{\partial p}{\partial u} \end{array} \right]^{-1} \left[ \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right] u + \left[ \begin{array}{c} \left( K_h^{-1} \right)_N \\ \frac{\partial B_{bc}^{out}}{\partial u} \\ -\frac{1}{\Delta T} \frac{\partial p}{\partial u} \end{array} \right]^{-1} \left[ \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right] p_{in} \quad (A.2)$$

$$= \frac{1}{\Delta T} J_{mod} u + B_{(i,N_z),sub} p_{in}$$

where $J_{mod}$ in equation (A.3), with $\psi = \left\{ \frac{v \cdot v}{2} - w c + \frac{c^2}{\gamma - 1} \right\}$, is the modification to $\bar{A}(x)$. The submatrix $B_{(i,N_z),sub}$ is the non-zero block of $B(x)$ for subsonic exit flow and will be defined in the next section.

$$J_{mod}(x) = \left[ \begin{array}{cccc}
-\frac{(\gamma - 1)v \cdot v}{2c^2} & \frac{(\gamma - 1)u}{c^2} & \frac{(\gamma - 1)w}{c^2} & -\frac{\gamma - 1}{c^2} \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)u^2}{c^2} & \frac{(\gamma - 1)u, w}{c^2} & -\frac{(\gamma - 1)u_r}{c^2} \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2}(w - c) & (\gamma - 1)u_r, w & -\frac{(\gamma - 1)u_r}{c^2} \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)w}{c^2} \\
-\frac{(\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & \frac{\gamma - 1}{c^2} \psi \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & \frac{\gamma - 1}{c^2} \psi \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & \frac{\gamma - 1}{c^2} \psi \\
\frac{\gamma - 1)v \cdot v}{2c^2} & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & -\frac{(\gamma - 1)v \cdot v}{2c^2} \psi & \frac{\gamma - 1}{c^2} \psi
\end{array} \right] \quad (A.3)$$
Combining the modifications at the entrance plane and those for subsonic exit flow produces the diagonal submatrices of $A(x)$:

$$A_{i,\text{sub}}(x) = \tilde{A}_{i,\text{sub}} + \begin{bmatrix} -\frac{1}{\Delta T} I & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{1}{\Delta T} J_{\text{mod}} \end{bmatrix} \quad (A.4)$$

When flow is supersonic at the exit plane, the state-dependent terms of $B_{bc,\text{sup}}^{\text{out}}(u, p_{\text{in}})$ are similar to $B_{bc}(x)$ and can be expressed as in equation (A.1). In this case, the first rows of $\tilde{A}_{i,\text{sup}}(x)$ are replaced with:

$$\begin{bmatrix} -\frac{1}{\Delta T} I & 0 & \ldots & 0 \end{bmatrix}$$

and the final rows are replaced with:

$$\begin{bmatrix} 0 & \ldots & 0 & -\frac{1}{\Delta T} I \end{bmatrix}$$

to form $A_{i,\text{sup}}$.

The state matrix $A(x)$ has a block-tridiagonal structure and can be constructed from the submatrices $A_{i,\text{sub}}$ and $A_{i,\text{sup}}$, depending on the flow conditions at the exit plane:

$$A(x) = \begin{bmatrix} A_1 & J_{F,1}^- & \zeta & \zeta & \ldots & \zeta \\ J_{F,2}^+ & A_2 & J_{F,2}^- & \zeta & \ldots & \zeta \\ \zeta & J_{F,3}^+ & A_3 & J_{F,3}^- & \ldots & \zeta \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \zeta & \ldots & \zeta & J_{F,N_r-1}^+ & A_{N_r-1} & J_{F,N_r-1}^- \\ \zeta & \ldots & \ldots & \zeta & J_{F,N_r}^+ & A_{N_r} \end{bmatrix} \quad (A.5)$$
A.2 Constructing the input matrix

The input matrix $B(x)$ has velocity-dependent blocks that can be found from the boundary condition functions $B_{bc}^{out}(u, p_{in})$:

\[
B_{bc,sub}^{out}(u, p_{in}) = p - p_{in}
\]

\[
B_{bc,sup}^{out}(u, p_{in}) = [\rho \rho u_r p - p_{in} \rho E]^T
\]

For subsonic flow at the exit plane, the block of $B(x)$ is:

\[
B_{(i,Nz),sub}(x) = \left[ (K_h^{-1})_N \frac{\partial B_{bc,sub}^{out}}{\partial u} \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\Delta T} \end{bmatrix}
\]

\[
\begin{array}{c}
\cdot \\
\frac{1}{c^2} \\
\frac{w_c}{c^2} \\
\frac{w - c}{c^2} \\
\frac{1}{c^2} \left\{ \frac{v \cdot v}{2} - w_c + \frac{c^2}{\gamma - 1} \right\}
\end{array}
\]

(A.6)
In the supersonic case, the contribution to $B(x)$ is:

$$B_{(i,N_z),sup} = -\frac{1}{\Delta T} \left( \frac{\partial B_{bc, sup}^{out}}{\partial \mathbf{u}} \right)^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$= \frac{1}{\Delta T} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{2-1}{2} \mathbf{v} \cdot \mathbf{v} & -(\gamma - 1)u_r & -(\gamma - 1)w & (\gamma - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix}_{x=\bar{x}}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{\Delta T} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{w(\gamma - 1)} \\ 0 \end{bmatrix}$$

(A.7)
The submatrices in equations (A.6) and (A.7) represent the non-zero block rows of $B(x)$, which has $N_z - 1$ block rows of zeros between the submatrices:

$$
B(x) = \begin{bmatrix}
Z \\
\vdots \\
Z \\
B_{1,N_z} \\
Z \\
\vdots \\
Z \\
B_2 \\
Z \\
\vdots \\
B_{N_r,N_z}
\end{bmatrix}
$$

(A.8)
Appendix B: One-dimensional Euler equation-based nonlinear model

The one-dimensional Euler equation-based model in equation (2.91) can be expressed in a form amenable for discretization through application of the chain rule:

\[
\frac{\partial \mathbf{u}}{\partial t} + J_h \frac{\partial \mathbf{u}}{\partial z} = J_k \mathbf{u}
\] (B.1)

This equation governs the flow at each point in the isolator grid and through discretization the state and input matrices of the nonlinear model can be defined, similar to the technique described in Section 2.2.9.

B.1 Spatially discretizing the Euler equations

Discretization of equation (B.1) allows for interconnections between grid points to be explicitly shown through the construction of a state vector formed by stacking the \( \mathbf{u}_k \) in order of increasing \( k \):

\[
x = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N_z} \end{bmatrix} \in \mathbb{R}^{3N_z} = \mathbb{R}^n
\] (B.2)
The governing equations can then be defined by stacking equation (B.1), applied at each grid point, in the same index order. During discretization, the flux splitting scheme of Steger and Warming [36] is applied to maintain stability in the model.

The split-Jacobians \( J_h^\pm \) are defined from the eigenvalues and eigenvectors of \( J_h \), \( J_h^\pm = K_h \Lambda^\pm K_h^{-1} \). The Jacobian has eigenvalues \( \lambda(J_h) = \{w, w + c, w - c\} \), from which \( \Lambda^\pm \) can be defined:

\[
\Lambda_h^\pm = \begin{bmatrix}
\frac{1}{2}(w \pm |w|) & 0 & 0 \\
0 & \frac{1}{2}(w + c \pm |w + c|) & 0 \\
0 & 0 & \frac{1}{2}(w - c \pm |w - c|)
\end{bmatrix}
\] (B.3)

The matrix \( K_h \) in equation (B.4) is constructed from the linearly-independent eigenvectors of \( J_h \) and has an inverse given in equation (B.5).

\[
K_h = \begin{bmatrix}
1 & \frac{\rho}{2c} & -\frac{\rho}{2c} \\
w & \frac{\rho}{2c}(w + c) & -\frac{\rho}{2c}(w - c) \\
\frac{w^2}{2} & \frac{\rho}{2c}(\frac{w^2}{2} + wc + \frac{c^2}{\gamma - 1}) & -\frac{\rho}{2c}(\frac{w^2}{2} - wc + \frac{c^2}{\gamma - 1})
\end{bmatrix}
\] (B.4)

\[
K_h^{-1} = \begin{bmatrix}
1 - \frac{\gamma - 1}{2} \frac{w^2}{c^2} & \frac{(\gamma - 1)w}{c^2} & -\frac{\gamma - 1}{c^2} \\
\frac{1}{\rho c}(\frac{\gamma - 1}{2} w^2 - wc) & \frac{1}{\rho c}(c - (\gamma - 1)w) & \frac{\gamma - 1}{\rho c} \\
-\frac{1}{\rho c}(\frac{\gamma - 1}{2} w^2 + wc) & \frac{1}{\rho c}(c + (\gamma - 1)w) & -\frac{\gamma - 1}{\rho c}
\end{bmatrix}
\] (B.5)

By definition, the split-Jacobians satisfy \( J_h = J_h^+ + J_h^- \).

For the interior points, the discretized Euler equations take the form:

\[
\frac{\partial \mathbf{u}}{\partial t} = -J_h^+ \frac{\mathbf{u}_k - \mathbf{u}_{k-1}}{z_k - z_{k-1}} - J_h^- \frac{\mathbf{u}_{k+1} - \mathbf{u}_{i,k}}{z_{k+1} - z_{i,k}} + J_k \mathbf{u}_k
\]

\[
= \frac{J_h^+}{z_k - z_{k-1}} \mathbf{u}_{k-1} - \frac{J_h^-}{z_{k+1} - z_k} \mathbf{u}_{k+1} + \left( \frac{J_h^-}{z_{k+1} - z_k} - \frac{J_h^+}{z_k - z_{k-1}} + J_k \right) \mathbf{u}_k
\] (B.6)
A boundary condition treatment similar to that developed for the two-dimensional Euler equation-based model, based on compatibility relations and time-differenced boundary conditions \cite{28,33,34}, is implemented for points on the edges of the grid. When \( k = N_z \), the boundary condition function is defined to include backpressure, the model input.

**B.1.1 Modifications for boundary points**

The flow at the entrance plane of the isolator is supersonic, but at the exit plane velocity-dependent boundary conditions are defined since flow may be subsonic or supersonic depending on the operating conditions of the engine. Since Chicatelli and Hartley do not treat the case of a downstream input with supersonic flow in their report, the treatment here is adapted from the case of an upstream input with supersonic flow \cite{28}.

In general, this boundary condition treatment involves transforming the model from state variables \( \mathbf{u} \) to characteristic variables \( \mathbf{v} \) and replacing physical characteristics with \( B_{bc} = 0 \). This transformation is defined by \( \partial \mathbf{v} / \partial t = K_h \partial \mathbf{u} / \partial t \). After the physical characteristics have been replaced with the boundary conditions, these functions are time-differenced to yield the expression:

\[
\frac{\partial \mathbf{u}}{\partial t} + \begin{bmatrix}
(K_h^{-1})_N & 0 \\
\frac{\partial B_{bc}}{\partial \mathbf{u}} & 0
\end{bmatrix}^{-1} \begin{bmatrix}
(K_h^{-1})_N \\
\frac{\partial \mathbf{u}}{\partial \mathbf{z}}
\end{bmatrix} J_h \frac{\partial \mathbf{u}}{\partial \mathbf{z}} = \begin{bmatrix}
(K_h^{-1})_N & 0 \\
\frac{\partial B_{bc}}{\partial \mathbf{u}} & 0
\end{bmatrix}^{-1} \begin{bmatrix}
(K_h^{-1})_N \\
\frac{\partial \mathbf{u}}{\partial \mathbf{z}}
\end{bmatrix} J_k \mathbf{u}
\]

\[
+ \begin{bmatrix}
(K_h^{-1})_N \\
\frac{\partial B_{bc}}{\partial \mathbf{u}}
\end{bmatrix}^{-1} \begin{bmatrix}
0 \\
-\frac{1}{\Delta T} B_{bc}
\end{bmatrix}
\]

\[(B.7)\]

where \((K_h^{-1})_N\) is the partition of \(K_h^{-1}\) associated with the numerical characteristics and \(\Delta T\) is the time-step for the simulation.
At the isolator entrance plane, \((K_h^{-1})_N\) is empty because there are no numerical characteristics for supersonic flow. Here, the boundary condition function is 

\[
B_{bc}^{in}(\mathbf{u}) = [\rho - \rho^* - (\rho w)^* (\rho E) - (\rho E)^*]^T,
\]

where \((\cdot)^*\) represents a ‘set-point’ for each state at \(k = 1\). This ‘set-point’ term was added to the boundary condition function because, during derivation of the one-dimensional Euler equation-based model, it was noted that the boundary conditions defined by Chicatelli and Hartley (and subsequently defined for the two-dimensional Euler equation-based PBM) were for the SPM, where the ‘set-points’ vanished (no perturbation). This modification helped avoid the undesirable decreases in pressure, density and velocity at the entrance and exit planes observed in Section 2.2.9.

Given \(B_{bc}^{in}(\mathbf{u})\), equation (B.7) simplifies to:

\[
\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\Delta T} (\partial B_{bc}^{in} / \partial \mathbf{u})^{-1} B_{bc}^{in}(\mathbf{u})
\]

which will be further simplified when the system is expressed in affine form.

Velocity-dependent boundary conditions are defined at the isolator exit plane. When the outflow is supersonic, the boundary condition is:

\[
B_{bc, sup}^{out}(\mathbf{u}, p_{in}) = \begin{bmatrix}
\rho - \rho^* \\
p - p_{in} \\
(\rho E) - (\rho E)^*
\end{bmatrix}
\]

since no numerical boundary conditions are assigned for supersonic flows. Following the treatment in Ref. 28, \(p_{in}\) is applied at the second characteristic. The governing equations for these grid points have the form in equation (B.8), with \(B_{bc, sup}^{out}(\mathbf{u}, p_{in})\) replacing \(B_{bc}^{in}(\mathbf{u})\).

For subsonic flow, there is only one physical characteristic and the boundary condition is \(B_{bc, sub}^{out}(\mathbf{u}, p_{in}) = p - p_{in}\). The two numerical characteristics in subsonic
flow are associated with the positive eigenvalues of $J_h$ and $(K_h^{-1})_N$ is the first two rows of $K_h^{-1}$. Defining the matrices $L$ and $L_B$ as:

\[
L = \left[ \frac{(K_h^{-1})_N}{\partial B^\text{out}_{bc,sub}/\partial u} \right]^{-1} \left[ \begin{array}{c} (K_h^{-1})_N \\ 0 \end{array} \right] = \left[ \begin{array}{ccc} 1 - \frac{(\gamma-1)w^2}{2c^2} & \frac{(\gamma-1)w}{c^2} & \frac{2-1}{c^2} \\ -\frac{(\gamma-1)w^2}{2c^2} & \frac{(\gamma-1)w}{c^2} & 1 - \frac{2-1}{c^2} \\ -\frac{(\gamma-1)w}{c} & \frac{w^2}{c^2} - wc + \frac{c^2}{\gamma-1} & -\frac{1}{c^2} \left( \frac{w^2}{c^2} - wc \right) \end{array} \right]
\]

\[
L_B = \left[ \frac{(K_h^{-1})_N}{\partial B^\text{out}_{bc,sub}/\partial u} \right]^{-1} \left[ \begin{array}{c} 0 \\ -\frac{1}{\Delta T} B^\text{out}_{bc,sub}(u,p_{in}) \end{array} \right] = \left[ \begin{array}{ccc} 1 - \frac{(\gamma-1)w^2}{2c^2} & \frac{(\gamma-1)w}{c^2} & \frac{2-1}{c^2} \\ \frac{1}{\rho c} \left( \frac{\gamma-1}{2} \right) w^2 - wc & \frac{1}{\rho c} (c - (\gamma-1)w) & \frac{2-1}{\rho c} \\ \frac{\gamma-1}{2} w^2 & -(\gamma-1)w & (\gamma - 1) \end{array} \right]^{-1} \left[ \begin{array}{c} 0 \\ \frac{1}{\Delta T} \left( p - p_{in} \right) \end{array} \right]
\]

allows for the governing equations to be expressed in the compact form:

\[
\frac{\partial u}{\partial t} = -L_J \frac{\partial u}{\partial z} + L_J u + L_B
\]

(B.11)

Here, the backwards-difference is used instead of flux-splitting to approximate the partial derivative with respect to $z$. Expansion of these nonlinear elements will yield a model with affine structure.

### B.2 Constructing intermediate state equations

Given $x$ in equation (B.2) and equations (B.6), (B.8), and (B.11), an intermediate form of the state equations can be constructed that is similar to the one described in section 2.2.7, with notation simplified through the definitions in equation (2.30).

Table B.1 shows the form of the governing equations applied at each grid point,
Table B.1: Explicit definition of the equations applied at each point in the CFD grid. Index $k$ corresponds to the interval in the streamwise direction to which each equation is applied, and the difference notation is given in equation (2.30). Note that where subscripts are dropped, it is assumed they are $k$.

<table>
<thead>
<tr>
<th>case</th>
<th>$k$</th>
<th>equation form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{in}^{u}}{\partial \mathbf{u}} \right)^{-1} B_{bc}^{in}(\mathbf{u})$</td>
</tr>
<tr>
<td>2</td>
<td>$\in (2, N_z - 1)$</td>
<td>$\frac{\partial \mathbf{u}}{\partial t} = -\frac{J_{kh}}{\Delta x_z} \Delta^{k} \mathbf{u} - \frac{J_{kh}}{\Delta y_z} \Delta^{k} \mathbf{u} + J_{k} \mathbf{u}$</td>
</tr>
<tr>
<td>3</td>
<td>$N_z$ (subsonic)</td>
<td>$\frac{\partial \mathbf{u}}{\partial t} = -\frac{LJ_{kh}}{\Delta x_z} \Delta^{k} \mathbf{u} + L J_{k} \mathbf{u} + L B(\mathbf{u}, p_{in})$</td>
</tr>
<tr>
<td>4</td>
<td>$N_z$ (supersonic)</td>
<td>$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{out}^{u}}{\partial \mathbf{u}} \right)^{-1} B_{bc, sup}^{out}(\mathbf{u}, p_{in})$</td>
</tr>
</tbody>
</table>

where boundary conditions are taken into consideration for cases 2 and 3 and case 3 is dependent on the velocity of the flow at the exit plane.

The expressions in Table B.1 can be stacked to form the state equations of the system:

$$
\begin{align*}
    k = 1 & \quad \frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\Delta T} \left( \frac{\partial B_{in}^{u}}{\partial \mathbf{u}} \right)^{-1} B_{bc}^{in}(\mathbf{u}) \\
    k = 2 & \quad \frac{\partial \mathbf{u}}{\partial t} = \frac{J_{kh}}{\Delta x_z} \mathbf{u}_1 - \frac{J_{kh}}{\Delta y_z} \mathbf{u}_3 + \left( J_k - \frac{J_{kh}}{\Delta x_z} + \frac{J_{kh}}{\Delta y_z} \right) \mathbf{u} \\
    & \vdots \\
    k = N_z & \quad \frac{\partial \mathbf{u}}{\partial t} = \frac{L J_{kh}}{\Delta x_z} \mathbf{u}_{N_z-1} + \left( L J_k - \frac{L J_{kh}}{\Delta x_z} \right) \mathbf{u} + L B(\mathbf{u}, p_{in})
\end{align*}
$$

(B.12)

from which $\bar{A}(x)$ and $\bar{B}(x, p_{in})$ can be constructed. Note subscripts dropped from $\mathbf{u}$ are assumed to be $k$, and that the expression given for $k = N_z$ is for subsonic exit flow. If the flow is supersonic at the exit plane, the expression in the last row of Table B.1 is used instead.

### B.2.1 Building the intermediate state matrix

The intermediate state matrix, $\bar{A}(x)$, can be constructed from the Jacobian $J_h$ and $J_k$ and expansion of the matrix $L$. In general, the matrix has a tridiagonal structure, where the block row corresponding to $k = N_z$ is dependent on the velocity of the flow at the exit plane.
flow at the exit plane: when it is subsonic, $\bar{A}(x)$ is given in equation (B.13), and in equation (B.14) when supersonic. Note that the dropped indicies on the Jacobians are the $k$ associated with that block row.

$$\bar{A}_{sub}(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & \ldots \\
\frac{J^+}{\Delta \phi z} & J_k - \frac{J^+}{\Delta \phi z} + \frac{J^-}{\Delta \phi z} & -\frac{J^-}{\Delta \phi z} & 0 & 0 & \ldots \\
0 & \frac{J^+}{\Delta \phi z} & J_k - \frac{J^+}{\Delta \phi z} + \frac{J^-}{\Delta \phi z} & -\frac{J^-}{\Delta \phi z} & 0 & \ldots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \frac{J^+}{\Delta \phi z} & J_k - \frac{J^+}{\Delta \phi z} + \frac{J^-}{\Delta \phi z} & -\frac{J^-}{\Delta \phi z} & 0 & \ldots \\
\vdots & 0 & \frac{LJ}{\Delta \phi z} & LJ_k - \frac{LJ}{\Delta \phi z} & \ldots & \ldots
\end{bmatrix} \quad (B.13)$$
\[ \mathbf{A}_{\text{sup}}(x) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \ldots \\ \frac{J_k^+}{\Delta t} - \frac{J_{k-}^-}{\Delta t} & J_k - \frac{J_{k-}^-}{\Delta t} + \frac{J_{k+}^+}{\Delta t} & - \frac{J_{k+}^+}{\Delta t} & 0 & 0 & \ldots \\ J_k \frac{J_k^+}{\Delta t} & - \frac{J_{k-}^-}{\Delta t} & - \frac{J_{k-}^-}{\Delta t} & 0 & 0 & \ldots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{bmatrix} \] (B.14)

These matrices are similar to the diagonal submatrices \( \mathbf{A}_{i,\text{sub}} \) (equation (2.38) or equation (2.39)) of the state matrix for the two-dimensional Euler equation-based PBM.

### B.2.2 Building the nonlinear boundary condition input vector

The velocity-dependent vector \( \mathbf{B}(x, p_m) \) is constructed by collecting the nonlinear terms resulting from the boundary condition treatment. Equations (B.15) and (B.16)
give the structure of $\bar{B}(x, p_{in})$ for subsonic and supersonic exit flow, respectively.

$$
\bar{B}_{sub}(x, p_{in}) = \begin{bmatrix}
-\frac{1}{\Delta T} \left( \frac{\partial \bar{B}_{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) \\
Z \\
\vdots \\
Z \\
L_B(u, p_{in})
\end{bmatrix}
$$  \hspace{1cm} (B.15)

$$
\bar{B}_{sup}(x, p_{in}) = \begin{bmatrix}
-\frac{1}{\Delta T} \left( \frac{\partial \bar{B}_{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) \\
Z \\
\vdots \\
Z \\
-\frac{1}{\Delta T} \left( \frac{\partial \bar{B}_{out}^{sup}}{\partial u} \right)^{-1} B_{bc,sup}^{out}(u, p_{in})
\end{bmatrix}
$$  \hspace{1cm} (B.16)

### B.3 Building the affine state equations

Application of Euler’s theorem of homogeneous functions to $\bar{B}(x, p_{in})$ allows for the isolator model to be expressed in an affine form by leading to construction of $A(x)$, $B(x)$ and $A^*(x)$. The ‘step-point’ term $A^*(x)x^*$ is introduced by the boundary condition functions and acts as a second input term in the model.

#### B.3.1 Constructing the state matrix

The state matrix, $A(x)$, can be construction by adding state-dependent terms from $\bar{B}(x, p_{in})$ to $\bar{A}(x)$, where modifications are made to block rows associated with $u_1$ and $u_{N_z}$. 

261
At the isolator entrance plane, the term modifying $\bar{A}(x)$ is found through expansion of the first block row of $\bar{B}(x, p_{in})$:

$$\begin{align*}
- \frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} B_{bc}^{in}(u) & = - \frac{1}{\Delta T} \left( \frac{\partial B_{bc}^{in}}{\partial u} \right)^{-1} (u - u^*) \\
& = - \frac{1}{\Delta T} I u + \frac{1}{\Delta T} I u^*
\end{align*} \tag{B.17}$$

Note that this term does not contribute to $B(x)$ because $B_{bc}^{in}(u) = u - u^*$ has no dependence on $p_{in}$.

For subsonic flow at the exit plane, the terms modifying $\bar{A}(x)$ are found through expansion of $L_B(u, p_{in})$ (in equation (B.10)) using Euler’s theorem of homogeneous functions:

$$L_B(u, p_{in}) = \left[ \left( K_h^{-1} \right)_N \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\Delta T} \frac{\partial p}{\partial u} \end{bmatrix} u + \left[ \left( K_h^{-1} \right)_N \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\Delta T} \end{bmatrix} p_{in} \tag{B.18}$$

$$= - \frac{1}{\Delta T} J_{mod} u + B_{N_{z,sub}p_{in}}$$

Here, $J_{mod}$ is given by

$$J_{mod} = \begin{bmatrix} \frac{(\gamma-1)w^2}{2c^2} & -\frac{(\gamma-1)w}{c^2} & \frac{\gamma-1}{c^2} \\ \frac{(\gamma-1)w^2}{2c^2} (w - c) & -\frac{(\gamma-1)w}{c^2} (w - c) & \frac{\gamma-1}{c^2} (w - c) \\ \frac{(\gamma-1)w^2}{2c^2} \psi & -\frac{(\gamma-1)w}{c^2} \psi & \frac{\gamma-1}{c^2} \psi \end{bmatrix} \tag{B.19}$$

where $\psi = \left\{ \frac{w^2}{2} - wc + \frac{c^2}{\gamma-1} \right\}$, and $B_{N_{z,sub}}$ is defined in the next section.
When the flow is subsonic at the exit plane, \( A(x) \) has the tridiagonal structure:

\[
A_{\text{sub}}(x) = \bar{A}_{\text{sub}}(x) + \begin{bmatrix}
-\frac{1}{\Delta T} I & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & -\frac{1}{\Delta T} J_{\text{mod}} 
\end{bmatrix}
\]  \tag{B.20}

and the similarities between this matrix and \( A_i(x) \) in equation (A.4) show the relationship between the two nonlinear models discussed in Chapter 2.

For supersonic flow at the exit plane, the state-dependent terms in the boundary condition function are expressed as in equation (B.17). (The input-dependent term will be discussed in the next section.) Like the modifications made for the state matrix of the two-dimensional Euler equation-based model, the first block row of \( \bar{A}_{\text{sup}}(x) \) is replaced with:

\[
\begin{bmatrix}
-\frac{1}{\Delta T} I & 0 & \ldots & 0 
\end{bmatrix}
\]

and the final block row is replaced with:

\[
\begin{bmatrix}
0 & 0 & \ldots & 0 & -\frac{1}{\Delta T} I 
\end{bmatrix}
\]

to form \( A_{\text{sup}}(x) \).
At the exit plane, when flow is supersonic, the boundary conditions contribute an additional term to the governing equations:

\[
\begin{align*}
A_{N_s}^*(x)u^* &= \frac{1}{\Delta T} \left( \frac{\partial B_{bc, sup}^{out}}{\partial u} \right)^{-1} \begin{bmatrix}
\rho^* \\
0 \\
(\rho E)^*
\end{bmatrix} \\
&= \frac{1}{\Delta T} \begin{bmatrix}
1 & 0 & 0 \\
\frac{w}{2} & -\frac{1}{w(\gamma-1)} & \frac{1}{w} \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\rho^* \\
0 \\
(\rho E)^*
\end{bmatrix}
\end{align*}
\]  
(B.21)

Together with the term introduced at \( k = 1 \), \( A_1^*(x)u^* = (\Delta T)^{-1}u^* \) (equation (B.17)), the ‘set-point’ term \( A^*(x)x^* \) is introduced to the model, where \( A^*(x) \) has the velocity-dependent structure given by:

\[
A_{sup}^*(x) = \frac{1}{\Delta T} \begin{bmatrix}
I & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ldots & 0 \\
0 & \ldots & 0 & \left[ \begin{array}{ccc}
1 & 0 & 0 \\
\frac{w}{2} & -\frac{1}{w(\gamma-1)} & \frac{1}{w} \\
0 & 0 & 1
\end{array} \right]
\end{bmatrix}
\]  
(B.22)

\[
A_{sub}^*(x) = \frac{1}{\Delta T} \begin{bmatrix}
I & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ldots & 0
\end{bmatrix}
\]
B.3.2 Constructing the input matrix

The boundary condition functions contribute to $B(x)$ terms dependent on the flow velocity at the exit plane, which are related to boundary conditions $B_{bc}(x, p_{in})$, repeated here for convenience:

\[
B_{bc, sub}^out(u, p_{in}) = p - p_{in}
\]
\[
B_{bc, sup}^out(u, p_{in}) = [\rho - \rho^* \ p - p_{in} (\rho E) - (\rho E)^*]^T
\]

The non-zero block of the $B$ matrix for the case of subsonic exit flow is given by:

\[
B_{Nz, sub}(x) = \left[(K_h^{-1})_N \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\Delta T} \end{bmatrix} = \frac{1}{\Delta T} \begin{bmatrix} \frac{1}{c^2} \\ \frac{w-c}{c^2} \\ \frac{1}{c^2} \left\{ \frac{w^2}{2} - wc + \frac{c^2}{\gamma-1} \right\} \end{bmatrix}
\] (B.23)
and by:

\[ B_{N_z, sup}(x) = -\frac{1}{\Delta T} \left( \frac{\partial B^{out}_{bc, sup}}{\partial u} \right)^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \]

\[ = \frac{1}{\Delta T} \begin{bmatrix} 1 & 0 & 0 \\ \frac{\gamma - 1}{2} w^2 & -(\gamma - 1)w & (\gamma - 1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \bigg|_{x = \bar{x}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \]  

(B.24)

\[ = \frac{1}{\Delta T} \begin{bmatrix} 0 \\ \frac{1}{w(\gamma - 1)} \\ 0 \end{bmatrix} \]

when the flow is supersonic. Equation (B.23) or (B.24) can be used to construct \( B(x) \), depending on the velocity of the exit flow:

\[ B_{sub}(x) = \begin{bmatrix} Z \\ \vdots \\ Z \\ B_{N_z, sub} \end{bmatrix} \]

\[ B_{sup}(x) = \begin{bmatrix} Z \\ \vdots \\ Z \\ B_{N_z, sup} \end{bmatrix} \]

(B.25)

**B.4 Constructing the nonlinear output function**

The nonlinear model of the isolator is completed by defining an output function that is affine with respect to the input, \( y = C(x)x + D_p_{in} \), where \( y \) is the pressure at
various locations along the isolator. The specific set of locations is not specified to maintain generality.

Because pressure $p = (\gamma - 1)\rho(e - \frac{u^2}{2})$ is a homogeneous function of degree one in $u$, it can be expressed as $p = (\partial p/\partial u)u$, where $(\partial p/\partial u)_k$ represents the non-zero blocks of $C(x)$, $C_k(u)$:

$$p_k = \left. \frac{\partial p}{\partial u} \right|_k = \left[ \frac{\gamma - 1}{2}w^2 - (\gamma - 1)w(\gamma - 1) \right] u_k$$

$$= C_k(u)u_k$$

(B.26)

where pressure measurements are made at the set of indices $\{k\}$. The output matrix has dimension $q$-by-$n$, where $q$ is the number of locations at which pressure is of interest. Stacking $p_k$ with an index order reflective of that used to construct $x$ yields:

$$y = \begin{bmatrix} p_{k_1} \\ p_{k_2} \\ \vdots \\ p_{k_{\text{end}}} \end{bmatrix}$$

(B.27)

where subscripts on $k$ indicate the element in the set of indices at which pressure is measured. For a specific output $q^*$, when $k \neq N_z$, $u_k$ is states $x_{3(k-1)+1}$ to $x_{3k}$ of the overall system, where the subscripts are the column range in row $q^*$ where $C_k(u)$ is inserted in $C(x)$, which has a block diagonal structure.

When $k = N_z$, input is fed directly through to the output, and the last row of $C(x)$ is zero while $D$ has a 1 in the same row. Consequently, $D$ is not state-dependent, but instead is a sparse matrix with a 1 in the last row if pressure at $k = N_z$ is of interest.
Given the definitions of $A(x)$, $B(x)$, $A^*(x)$, $C(x)$, and $D$, the system has the affine form:

$$\dot{x} = A(x)x + B(x)p_{in} + A^*(x)x^*$$
$$y = C(x)x + Dp_{in}$$

(B.28)

where $x \in \mathbb{R}^{3N_z}$, $u \in \mathbb{R}$, and $y \in \mathbb{R}^q$ and the system matrices have the following dimensions:

$$A(x) \in \mathbb{R}^{3N_z \times 3N_z} \quad B(x) \in \mathbb{R}^{3N_z \times 1} \quad A^*(x) \in \mathbb{R}^{3N_z \times 3N_z}$$

$$C(x) \in \mathbb{R}^{q \times 3N_z} \quad D \in \mathbb{R}^{q \times 1}$$

which are compatible with the state, input, and output dimensions.

### B.5 Damping Modification

Simulation of the nonlinear model with an input representing an increase of back-pressure revealed that the shock remained stationary instead of propagating upstream (this will be shown in section 2.3.2). This artifact is rectified in CFD simulations of the Euler equations through the inclusion of artificial dissipation (also referred to as artificial viscosity or damping) in the model [34]. The damping scheme described in section 2.2.9 is adapted here by neglecting terms related to finite-differences taken in the radial direction (index $i$).

Following the notation of Swanson and Turkel [40], the damping scheme for a grid point $k$ is given by:

$$(D_z^2 - D_z^4)u_k = \{\nabla_z [\lambda_{k+0.5} \epsilon_{k+0.5}^2 \Delta_z] - \nabla_z [\lambda_{k+0.5} \epsilon_{k+0.5}^4 \Delta_z \nabla_z \Delta_z]\} u_k$$

(B.29)
where the backwards-difference operator is indicated by $\nabla$ and the forward-difference by $\Delta$, and scaling factors are:

$$\lambda_{k+0.5} = \frac{1}{2} [\lambda_{z,k+1} + \lambda_{z,k}]$$

$$\epsilon^2_{k+0.5} = \kappa^2 \max \{\nu_k, \nu_{k+1}\} \quad \text{(B.30)}$$

$$\epsilon^4_{k+0.5} = \max \{0, \kappa^4 - \epsilon^2_{k+0.5}\}$$

Scaling factors $\lambda_{k*}$ are related to the maximum eigenvalues of $J_h$ at the indicated grid points ($\lambda_z = |w| + c$). Although $\kappa^{2,4}$ are nominally equal to $1/4$ and $1/256$ (as suggested in Ref. 42), additional scaling is introduced to achieve accurate simulation results. The most influential term in equation (B.30) is $\nu_k$:

$$\nu_k = \frac{|p_{k+1} - 2p_k + p_{k-1}|}{|p_{k+1} + 2p_k + p_{k-1}|} \quad \text{(B.31)}$$

which is related to the pressure gradient in the flow.

The state equations can be modified to include damping matrices, $P^2(x)$ and $P^4(x)$, that are constructed from these nonlinear damping terms:

$$\dot{x} = A(x)x + B(x)u + A^*(x)x^* + P^2(x)x - P^4(x)x$$

$$= (A(x) + P^2(x) - P^4(x))x + B(x)u + A^*(x)x^* \quad \text{(B.32)}$$

$$= \hat{A}(x)x + B(x)u + A^*(x)x^*$$

The effect of including damping in the model can be studied in simulation of this model.
B.5.1 Constructing the Second-Order Damping Matrix

Because $\epsilon^2$, in equation (B.30), depends on a four-point stencil, a boundary condition treatment is needed for points at the edges ($k = 1, Nz$) and just inside the edges ($k = 2, Nz - 1$) of the grid.

For a general grid point $k$, the second-order damping terms are:

$$D_z^2 u_k = \{ \nabla_z \left[ \lambda_{k+0.5} \epsilon_{k+0.5}^2 \Delta_z \right] \} u_k$$

$$= \lambda_{k+0.5} \epsilon_{k+0.5}^2 u_{k+1} + \lambda_{k-0.5} \epsilon_{k-0.5}^2 u_{k-1} - \left[ \lambda_{k+0.5} \epsilon_{k+0.5}^2 + \lambda_{k-0.5} \epsilon_{k-0.5}^2 \right] u_k \quad (B.33)$$

$$= \alpha_{k+1} u_{k+1} + \alpha_{k-1} u_{k-1} - \alpha^k u_k$$

where $\alpha_{k\pm1} = \lambda_{k\pm0.5} \epsilon_{k\pm0.5}^2$ and $\alpha^k = \alpha_{k+1} + \alpha_{k-1}$ for a given $k$.

Boundary Condition Treatment at Grid Edge

This boundary treatment is adapted from that implemented in the damping scheme for the two-dimensional Euler equation-based model in section 2.2.9. There is no damping at the entrance plane ($\alpha_1 = \alpha_2 = 0$) because there are only physical boundary conditions for supersonic flow and, for simplicity, this same scheme was implemented at the exit plane ($\alpha - Nz - 1 = \alpha_{Nz} = 0$), incurring possible local error in the model.

Boundary Condition Treatment at First Interior Point

A similar boundary condition is implemented for points with $k = 2$ and $k - Nz - 1$, although no coefficients are set to zero since all the finite-differences can be calculated.
Along these boundaries, $\alpha_{k\pm 1}^*$ depends on the grid location:

$$
\alpha_{k+1}^* = \begin{cases} 
\alpha_{k+1} & \text{for } k = 2 \\
\lambda_{k+0.5} \kappa^2 \nu_k & \text{for } k = N_z - 1 
\end{cases}
$$

$$
\alpha_{k-1}^* = \begin{cases} 
\lambda_{k-0.5} \kappa^2 \nu_k & \text{for } k = 2 \\
\alpha_{k-1} & \text{for } k = N_z - 1 
\end{cases}
$$

(B.34)

Like the treatment described in section 2.2.9, the maximum function is not evaluated when the stencil for calculating $\nu_{k\pm 1}$ extends beyond the grid. For example, when $k = 2$, evaluation of the maximum is replaced with direct calculation of $\nu_k$. This boundary condition treatment does not affect the definition of $\alpha_k$.

The block tridiagonal matrix $P^2(x)$ in equation (B.35) is also a banded matrix, with three bands distributed about the diagonal.

$$
P^2(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & \ldots & 0 & 0 \\
\alpha_{N_z-2}^* & -\alpha_{N_z-1}^* & \alpha_{N_z} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \alpha_{N_z-2} & -\alpha_{N_z-1} & \alpha_{N_z} \\
0 & 0 & 0 & \ldots & 0 & 0 & 0
\end{bmatrix}
$$

(B.35)

Note that $\alpha_k^*$ is defined in equation (B.34) and $\alpha_k$ is the sum of the coefficients in the block row associated with grid point $k$. 
B.5.2 Building the Fourth-Order Damping Matrix

The fourth-order damping terms in equation (B.29) can be expanded as:

\[ D_z^4 u_k = \lambda_{k+0.5} \epsilon_{k+0.5}^4 (u_{k+2} - 3u_{k+1} + 3u_k - u_{k-1}) \]
\[ - \lambda_{k-0.5} \epsilon_{k-0.5}^4 (u_{k+1} - 3u_k + 3u_{k-1} - u_{k-2}) \]
\[ = \lambda_{k+0.5} \epsilon_{k+0.5}^4 u_{k+2} - (\lambda_{k-0.5} \epsilon_{k-0.5}^4 + 3\lambda_{k+0.5} \epsilon_{k+0.5}^4) u_{k+1} \]
\[ - (3\lambda_{k-0.5} \epsilon_{k-0.5}^4 + \lambda_{k+0.5} \epsilon_{k+0.5}^4) u_{k-1} \]
\[ + \lambda_{k-0.5} \epsilon_{k-0.5}^4 u_{k-2} + 3(\lambda_{k+0.5} \epsilon_{k+0.5}^4 + \lambda_{k-0.5} \epsilon_{k-0.5}^4) u_k \]
\[ = \beta_{k+1} u_{k+2} - (3\beta_{k+1} + \beta_{k-1}) u_{k+1} - (\beta_{k+1} + 3\beta_{k-1}) u_{k-1} \]
\[ + \beta_{k-1} u_{k-2} + \beta^k u_k \]

where \( \beta_{k\pm} = \lambda_{k\pm0.5} \epsilon_{k\pm0.5}^4 \) and \( \beta^k \) is the sum of the coefficients on \( u_{k\pm2} \) and \( u_{k\pm1} \).

Equation (B.36) is more complex than equation (B.33) because of the extra differencing operations on \( u \). In constructing \( P^4(x) \) from equation (B.36), boundary conditions at \( k = 1, 2, N_z - 1, N_z \) must be taken into consideration.

Boundary Condition Treatment at Grid Edge

Similar to the treatment in section 2.2.9, a three-point central difference replaces the five-point central differences for grid points with \( k = 1 \) and \( k = N_z \):

\[ -D_z^4 u_k = - \left\{ \nabla_z \left[ \lambda_{k+0.5} \epsilon_{k+0.5}^4 \Delta_z \right] \right\} u_k \]
\[ D_z^4 u_k = \lambda_{k+0.5} \epsilon_{k+0.5}^4 (u_{k+1} - u_k) - \lambda_{k-0.5} \epsilon_{k-0.5}^4 (u_k - u_{k-1}) \]
\[ = \lambda_{k+0.5} \epsilon_{k+0.5}^4 u_{k+1} + \lambda_{k-0.5} \epsilon_{k-0.5}^4 u_{k-1} \]
\[ - (\lambda_{k+0.5} \epsilon_{k+0.5}^4 + \lambda_{k-0.5} \epsilon_{k-0.5}^4) u_k \]
\[ = \beta^*_{k+1} u_{k+1} - \beta^*_{k-1} u_{k-1} + \beta^k u_k \]
Note that $\beta^k = \beta_{k+1}^* + \beta_{k-1}^*$, the sum of the coefficients of each $u_{k\pm 1}$, which preserves the definition of $\beta^k$ at the grid edge. Because this expression is similar to the second-order dissipation term, the boundary condition treatment discussed for the second-order damping term is extended here.

Equation (B.37) is simplified for $k = 1, N_z$, because damping is not implemented at the isolator entrance and exit planes: $\beta_1 = \beta_2 = \beta_{N_z-1} = \beta_{N_z} = 0$.

**Boundary Condition Treatment at First Interior Point**

For grid points with $k = 2, N_z - 1$, equation (B.37) holds with the following definitions:

\[
\begin{align*}
\beta_{k+1}^* &= \begin{cases} 
\beta_{k+1} & \text{for } k = 2 \\
\lambda_{k+0.5} \max\{0, \kappa^4 - \kappa^2 \nu_k\} & \text{for } k = N_z - 1
\end{cases} \\
\beta_{k-1}^* &= \begin{cases} 
\lambda_{k-0.5} \max\{0, \kappa^4 - \kappa^2 \nu_k\} & \text{for } k = 2 \\
\beta_{k-1} & \text{for } k = N_z - 1
\end{cases}
\end{align*}
\]

(B.38)

where the modifications from the second-order dissipation terms to the fourth-order terms through the arguments of the maximum function. The definition of $\beta^k$ does not change for these point.

Because of the the extended stencil, $P^4(x)$ is more complex than $P^2(x)$ as it has a block pentadiagonal structure and is a banded matrix with five band distributed
about the diagonal:

\[
P^4(x) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & \ldots \\
\beta_{*1} & -\beta^2 & \beta_3 & 0 & 0 & \ldots \\
\beta_2 & -(3\beta_2 + \beta_4) & -\beta^3 & -(\beta_2 + 3\beta_4) & \beta_4 & \ldots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
0 & 0 & 0 & 0 & 0 & \ldots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \cdots \\
\cdots & \beta_{N_z - 3} & -(3\beta_{N_z - 3} + \beta_{N_z - 1}) & -\beta_{N_z - 2} & -(\beta_{N_z - 3} + 3\beta_{N_z - 1}) & \beta_{N_z - 1} & \ldots \\
\cdots & 0 & 0 & \beta_{N_z - 2} & -\beta_{N_z - 1} & \beta_{*N_z} & \ldots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots 
\end{bmatrix}
\]  \tag{B.39}

Unlike \(\alpha^k\), \(\beta^k\) is not always the sum of each damping coefficient \(\beta_{k \pm 1}\) because of the presence of the fourth-derivative terms:

- \(k = 1, N_z\): \(\beta^k = 0\)
- \(k = 2, N_z - 1\): \(\beta^k = \beta_{k-1} + \beta_{k+1}\)
- \(k \in (2N_z - 1)\): \(\beta^k = 3(\beta_{k+1} + \beta_{k-1})\)

This can be summarized as “\(\beta^k = \) the sum of the coefficients on \(u_{k \pm 2}\) and \(u_{k \pm 1}\),” as emphasized throughout the discussion.

It was found that the nominal values \(\kappa^2 = 1/4\) and \(\kappa^4 = 1/256\) did not scale the damping parameters enough to affect the behavior of the model, as discussed in section 2.2.9 for the two-dimensional Euler equation-based model. To compensate for this, each coefficient \(\alpha_{k \pm 1}\) and \(\beta_{k \pm 1}\) was multiplied by a tunable scaling factor, as addressed in section 2.3.2.