STABILITY ASSESSMENT OF NONLINEAR SYSTEMS USING THE LYAPUNOV EXPONENT

A Thesis Presented to

The Faculty of

Fritz J. and Delores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirements for the Degree

Masters of Science

by

Eric A. Best

June, 2003

OHIO UNIVERSITY
LIBRARY
Acknowledgments

Reaching this milestone is the culmination of many years of hard work. I could not have achieved this without the support, guidance and knowledge of many special people:

Dr. J. Jim Zhu, my thesis advisor, whose expertise made this project a success and for giving me the opportunity to work for and learn from him; Dr. Douglas Lawrence, Dr. Michael Braasch and Dr. Jae Lew for giving their time to review my thesis; To Dr. Jim Rankin for providing me the opportunity to conduct research as an undergraduate student and serving as a true mentor; To Warren Adams, Bob Jones and Greg Dukeman from NASA Marshal Space Flight Center for support using MAVERIC; Ms. Marjorie Burkhart, McKinley Senior High School, for instilling in me an appreciation for mathematics; Mr. Jack Lang, McKinley Senior High School, for believing in me and helping me believe in myself.

On a more personal note, I would like to thank the following people for their love and support through the years:

I would like to thank Michele Seymour for her love, support, patience and understanding; Thank you to my family for all the love and support; A very special thanks to my parents who have always supported me and encouraged me to succeed. Their love and support never goes unappreciated.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>Symbols and Abbreviations</td>
<td>viii</td>
</tr>
<tr>
<td>Chapter 1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Chapter 2 Lyapunov's First Method</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Stability of Motion</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Equilibrium State of Motion</td>
<td>6</td>
</tr>
<tr>
<td>2.2.2 Stability</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Lyapunov Exponent Stability Assessment for NL Systems</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Lyapunov's First Method</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Definition of the Lyapunov Exponent</td>
<td>16</td>
</tr>
<tr>
<td>2.3.3 Lyapunov Exponent Behavior</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 3 Departure Index - An Instability Indicator</td>
<td>22</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>22</td>
</tr>
<tr>
<td>3.2 Concepts/Principles</td>
<td>23</td>
</tr>
<tr>
<td>3.3 Practical Issues and Solutions</td>
<td>25</td>
</tr>
<tr>
<td>3.3.1 Dead Band</td>
<td>29</td>
</tr>
<tr>
<td>3.3.2 Low Pass Filter</td>
<td>30</td>
</tr>
<tr>
<td>3.3.3 Hilbert Transformation</td>
<td>31</td>
</tr>
<tr>
<td>3.4 Implementation</td>
<td>34</td>
</tr>
<tr>
<td>3.4.1 Discrete Time</td>
<td>34</td>
</tr>
<tr>
<td>3.4.2 Continuous Time</td>
<td>36</td>
</tr>
<tr>
<td>Chapter 4 Results</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>39</td>
</tr>
<tr>
<td>4.2 DI Accuracy and Promptness</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Delay and Gain Margin Tests</td>
<td>43</td>
</tr>
<tr>
<td>4.3 Short DI Window Size Tests</td>
<td>45</td>
</tr>
<tr>
<td>Chapter 5 Summary</td>
<td>52</td>
</tr>
<tr>
<td>5.1 Conclusions</td>
<td>52</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>54</td>
</tr>
<tr>
<td>References</td>
<td>56</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1 Bandwidth and Damping Ratio values for Tests 1 and 2</td>
<td>40</td>
</tr>
<tr>
<td>4.2.2 Error threshold values for inner and outer loop ascent control</td>
<td>41</td>
</tr>
<tr>
<td>4.2.3 SDI and LDI thresholds for each case</td>
<td>41</td>
</tr>
<tr>
<td>4.3.1 Failed runs in Cases 1-5 (Normal Termination)</td>
<td>44</td>
</tr>
<tr>
<td>4.3.2 Failed runs in Cases 1-5 (Core Dumps)</td>
<td>44</td>
</tr>
<tr>
<td>4.4.1 Summary for number of SDI detections prior to tracking error detections</td>
<td>46</td>
</tr>
<tr>
<td>4.4.2 SDI threshold values for Test 3</td>
<td>47</td>
</tr>
<tr>
<td>4.4.3 Number of false alarms by the SDI</td>
<td>50</td>
</tr>
<tr>
<td>4.4.4 SDI threshold statistics</td>
<td>51</td>
</tr>
<tr>
<td>Figure Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.2.1 Domain of attraction</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Convergence rate of an attractor</td>
<td>10</td>
</tr>
<tr>
<td>2.2.3 Lyapunov stability of an equilibrium</td>
<td>12</td>
</tr>
<tr>
<td>2.2.4 Domain of stability</td>
<td>13</td>
</tr>
<tr>
<td>2.2.5 Stability of a nominal trajectory</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Lyapunov Exponent Behavior</td>
<td>20</td>
</tr>
<tr>
<td>3.2.1 Entire Integrated Guidance and Control system</td>
<td>23</td>
</tr>
<tr>
<td>3.3.1 SDI block diagram with disturbances</td>
<td>26</td>
</tr>
<tr>
<td>3.3.2 Inner-loop yaw LDI, SDI and tracking error before low pass filter and Dead Band introduced</td>
<td>27</td>
</tr>
<tr>
<td>3.3.3 Outer-loop yaw LDI, SDI and tracking error after low pass filter and Dead Band introduced</td>
<td>28</td>
</tr>
<tr>
<td>3.3.4 Graphical representation of Dead Band logic</td>
<td>29</td>
</tr>
<tr>
<td>3.4.1 SDI block diagram with other subsystems</td>
<td>35</td>
</tr>
<tr>
<td>3.4.2 LDI, SDI and tracking error for outer-loop roll channel</td>
<td>36</td>
</tr>
<tr>
<td>3.4.3 Block diagram of SDI implementation in IDOS</td>
<td>37</td>
</tr>
<tr>
<td>3.4.4 Block diagram of Short DI integral</td>
<td>37</td>
</tr>
<tr>
<td>3.4.5 Outer-loop pitch channel SDI and tracking error in IDOS</td>
<td>38</td>
</tr>
<tr>
<td>4.4.1 Inner-loop tracking errors and SDIs for a false alarm run</td>
<td>49</td>
</tr>
</tbody>
</table>
Symbols and Abbreviations

2GRLV.................................................. 2nd Generation Reusable Launch Vehicle
6-DOF................................................. 6-Degree of Freedom
AG&C............................................... Advanced Guidance and Control
DI..................................................... Departure Index
DOA................................................... Domain of Attraction
HHT.................................................. Hilbert-Huang Transformation
HT.................................................... Hilbert Transformation
IDOS............................................... Integrated Development Operations System
IG&C................................................ Integrated Guidance and Control
ITAR................................................ International Traffic in Arms Regulations
LDI.................................................... Long Departure Index
LTI..................................................... Linear Time-Invariant
LTV................................................... Linear Time-Varying
MAVERIC........................................ Marshall Aerospace Vehicle Representation In C
MSFC................................................ Marshall Space Flight Center
NASA............................................. National Aeronautics and Space Administration
NLTI................................................. Nonlinear Time-Invariant
NLTV.............................................. Nonlinear Time-Varying
RLV................................................ Reusable Launch Vehicle
SDI................................................... Short Departure Index
SLIM............................................... Space Launch Initiative MAVERIC
s.t................................................... such that
w.r.t............................................... with respect to
∀.................................................... for all
∃.................................................... there exists
end of a definition
end of a theorem
set of real numbers
matrix transpose
Chapter 1 Introduction

In June of 2001, a contract for the National Aeronautics and Space Administration's (NASA's) Space Launch Initiative (SLI) program was awarded to Ohio University. The main objectives of the SLI program were to increase efficiency, increase safety and decrease the cost of low earth orbit space travel. The main product of the SLI program was to be a Second Generation Reusable Launch Vehicle (2GRLV) which would replace the Space Shuttle with increased reliability and lower operational costs. Ohio University was tasked with the development of a fully autonomous Integrated Guidance and Control (IG&C) system which would enable the 2GRLV to meet the requirements set forth for the SLI program.

Such a vehicle would prove to be enormously complex and require advanced methods to monitor the vehicle's performance. For the purposes of this thesis, the nominal guidance trajectory is defined as \( \bar{x}(t) \). The vehicle's actual trajectory is defined as \( \bar{E}(t) \) and the tracking error is defined as \( x(t) = \bar{x}(t) - \bar{E}(t) \). A nonlinear control technique known as Control by Trajectory Linearization was implemented as the control system for the 2GRLV. A detailed description of this method can be obtained from Zhu [18]. In addition, the tracking error of the control system can also be referred to as the operating error. Ideally, it is desired the vehicle follow the commanded guidance trajectory perfectly which would result in a zero tracking error (i.e. \( x(t) = \bar{x}(t) - \bar{E}(t) = 0 \)).

The focus of this research is the assessment of the stability of the motion of the vehicle which leads to the complicated notion of stability of motion. In this case, the stability of the guidance trajectory is assessed. Stability as defined by Aleksandr Lyapunov states that given initial time \( t_0 \), if the trajectory starts close to a stable equilibrium point, it will remain close for all time \( t \rightarrow \infty \). In terms of this thesis, a guidance trajectory \( \bar{x}(t) \) will be considered stable if the actual trajectory \( \bar{E}(t) \) remains close to \( \bar{x}(t) \). This would result in a small tracking error \( x(t) \).
The main result of this thesis is the use of Lyapunov's Indirect Method (first method) as a stability assessment approach. Lyapunov's Indirect Method involves the solution of a given nonlinear equation at a given initial condition within the envelope of operation. This is unlike the more well known Lyapunov's Direct Method (second method) in which a suitable Lyapunov function is first determined for the nonlinear system, allowing stability to be assessed "directly" without solving for the solution. Finding a suitable function can be very difficult if not impossible, making the second method unsuitable for this application.

Since the first method is used, in order to show stability of a system's motion, the solution to the system's governing differential equation would have to be calculated for every possible initial condition. This could be extremely computationally expensive if not impossible. The Lyapunov exponent indicates whether the actual trajectory departs from the nominal trajectory, thus it is called a Departure Index (DI). For this reason the DI will be referred in this thesis as an instability detection technique rather than a stability detection technique. All that is required is one unstable case in order to classify the guidance trajectory for a given initial condition as unstable. Therefore, the developed technique is used as an instability detection algorithm.

The approach used to develop and demonstrate the proposed algorithm was to implement the DI in MAVERIC (Marshal Aerospace Vehicle Representation in C) and SLIM (Space Launch Initiative MAVERIC), which are high fidelity 6-DOF (Degree of Freedom) simulation packages that include the vehicle model for the X-33 experimental launch vehicle. The DI has also been implemented in a SIMULINK based high fidelity 6-DOF RLV (Reusable Launch Vehicle) simulation package known as IDOS (Integrated Development Operation System). The simulation packages include all necessary components for simulating the flight of a real launch vehicle. Among these components are the guidance, navigation and control systems. The DI is added to the control system as
a monitor of the controller's tracking error and has no effect on the controller's operation. A set of tests were developed to show the DI's timeliness and accuracy in indicating onset of instability as opposed to the analysis of the tracking error alone. Also, the relation between Gain and Phase margins, which is a well known approach for assessing a controller's robustness for LTI systems, will be extended to nonlinear time-varying systems as Gain and Delay margins. These tests were conducted using SLIM by running numerous Monte Carlo dispersion tests.

The use of Monte Carlo dispersion tests is not only time-consuming and costly, it is often difficult to gauge the robustness, or stability margins of the system from the dispersion test results. For this reason, it is necessary to develop a technique that would eliminate the need for a large number of dispersion tests. Currently, the Lyapunov exponent is being used with Monte Carlo tests to provide more useful meaning to the test results. The intent is that the development of the Lyapunov exponent, for this application, will eventually prove to be an accurate analytic method for assessing stability, thus only requiring one or a few simulations to be analyzed and not several hundred or thousand.

A number of references have been reviewed to ensure that the idea of using the Lyapunov exponent for stability assessment has not been previously explored. This method has proven to be very effective in a number of other fields, including the stock market, biotechnology, the study of chaotic systems and the study of the stability of the universe. Very few papers addressed the use of this method in reference to control systems, but this thesis will show the method is suitable for this application. In a number of papers ([6], [13], and [16]), a similar method of calculating the Lyapunov exponent is presented. Also, in [1] and [14], versions of a short-term Lyapunov exponent are explored.

This thesis contains four chapters. The first chapter presents a rather rigorous discussion of stability of motion and stability of equilibria as well as the theoretical
background for the Lyapunov exponent. The second chapter covers the implementation of the DI, the issues that arose through development and solutions to those issues. The third chapter describes the tests used to verify the DI as a capable stability assessment approach and presents the results. The fourth and final chapter is the summary, which includes conclusions and proposed future work.
Chapter 2 Lyapunov's First Method

2.1 Introduction

This chapter presents a rather rigorous discussion of stability of motion and equilibria as well as the theoretical background for the Lyapunov exponent. This thesis is concerned with nonautonomous systems and the equation for such a system is presented first (Section 2.2). Second, the definition of an equilibrium is included, along with the properties of attractiveness and uniform attractiveness (Section 2.2.1). In addition, the notion of convergence rate is presented, which finally leads to a discussion on the possible forms of stability of a unique solution (Section 2.2.2). Each concept is presented in a theoretical manner and then explained as they apply directly to this research. Figures are presented to illustrate these key concepts as they are important in understanding the Lyapunov exponent.

Lyapunov's First Method and the Lyapunov exponent are then defined for linear systems (Sections 2.3.1 - 2.3.2). A discussion of the behavior of the Lyapunov exponent is presented to illustrate how it is used in this research (Section 2.3.3). Finally, the notion of a short-term Lyapunov exponent is presented and briefly explained. Further discussion of the short-term exponent is presented throughout the thesis as it is the fundamental development of this research.

2.2 Stability of Motion

This thesis is mainly focused on the stability of the motion of dynamical systems that are time varying and highly nonlinear. These qualities are inherit in the nature of any current spacecraft as its weight changes at a rapid pace, travels at very high speeds and through numerous atmospheric conditions in a short time interval. To further complicate matters, the analysis of stability not only considers the designed system, but also that of
any external disturbances applied to the system. This makes the analysis of the system itself very difficult.

For the purposes of this thesis, the stability of solutions to nonautonomous systems of the following form

\[ \dot{x} = f(t, x), \]

will be considered, where \( x \) is the current operating state and \( t \) is time. The stability of motion in the sense of Lyapunov is concerned with the ability of an equilibrium of a nonlinear system to confine or attract the motion in its neighborhood. This leads to the definition of an equilibrium state of motion and how it is used in this research.

### 2.2.1 Equilibrium State of Motion

In this research, the desired operating point or nominal guidance trajectory for the closed-loop tracking controller is an equilibrium state denoted by \( x_e \). This leads to the following formal definition of an equilibrium state.

**Definition 2.2.1** A vector \( x_e \in \mathbb{R}^n \) is called an equilibrium state of a nonlinear dynamic system

\[ \dot{x} = f(t, x), \quad x(t_0) = x_0, \quad t \geq t_0, \]

if \( f(t, x_e) \equiv 0 \). Furthermore, an equilibrium \( x_e \) is said to be isolated if there exists a \( \delta > 0 \) such that for any other equilibrium point \( \tilde{x}_e \) of 2.2.2, \( \|x_e - \tilde{x}_e\| > \delta \).

In particular, if \( x \) in Equation 2.2.2 represents the tracking error of the nonlinear tracking controller, then the condition \( x_e = 0 \) would represent the perfect equilibrium (operating) state of the controller. That is, the closed-loop trajectory is following the commanded trajectory perfectly.
It should be noted that for stable linear systems which can have at most one isolated equilibrium, a nonlinear system may have a finite or even an infinite number of equilibria. Moreover, the number or even location of the equilibria may change under parametric perturbations, known as the bifurcation phenomenon.

For simplicity, this thesis will consider the equilibrium state to be the origin, as a simple coordinate transformation can be applied to make this assumption valid. Because of this assumption, \( x_e = 0 \) will be designated as an isolated equilibrium of Equation 2.2.2 and Equation 2.2.2 has the unique solution \( x(t) = s(t, t_0, x_0) \) for all \( t_0 \geq T_0 \) and \( x_0 \in \Omega \subset \mathbb{R}^n \), where \( \Omega \) is a simply connected region containing the origin (the operating envelope). With this, the stability concepts of Lyapunov for equilibria and trajectories are defined as follows.

An equilibrium can be attractive, uniformly attractive or even not attractive. The definitions for attractive and uniformly attractive equilibria are presented here.

**Definition 2.2.2** An equilibrium \( x_e = 0 \) of Equation 2.2.1 is said to be attractive if for all \( t_0 \geq T_0 \), there exists an \( \eta = \eta(t_0) > 0 \) such that \( \|x_0\| < \eta(t_0) \) implies \( s(t, t_0, x_0) \to 0 \) as \( (t - t_0) \to \infty \).

**Definition 2.2.3** An equilibrium \( x_e = 0 \) of Equation 2.2.1 is said to be uniformly attractive if for all \( t_0 \geq T_0 \), there exists an \( \eta > 0 \), independent of \( t_0 \), such that \( \|x_0\| < \eta \) implies \( s(t, t_0, x_0) \to 0 \) as \( (t - t_0) \to \infty \), uniformly in \( t_0, x_0 \), i.e. given an \( \epsilon > 0 \), there exists a \( T = T(\epsilon) > 0 \), independent of \( t_0, x_0 \), such that \( \|s(t, t_0, x_0)\| < \epsilon \) whenever \( t > t_0 + T \).

In simple terms, an equilibrium is attractive if all solutions nearby, i.e. originated within a bound \( \eta \) at \( t_0 \), eventually converge to the equilibrium as \( t \to \infty \). An equilibrium is considered uniformly attractive if all solutions nearby eventually converge to the
equilibrium as $t \to \infty$, however, the bound $\eta$ does not vary with initial time, therefore, the initial time need not be taken into consideration. These concepts are demonstrated graphically in Figure 2.2.1 (a-b)
When considering the attractiveness of an equilibrium, it is also necessary to consider the rate of convergence of the trajectory. Therefore, a discussion of rate of convergence is in order. The rate of convergence is an indication of the amount of time it takes for a solution to converge to the domain of attraction (DOA). The convergence rate can be uniform or non-uniform. If the convergence rate is uniform, it would take a solution the same time $T$ to converge to the DOA that originates at the point $x(t_0)$ as it would take a point $\bar{x}(t_1)$ to converge to the DOA. This means the point $x(t_0)$ would converge to the DOA at the point $x(t_0 + T)$ and the point $\bar{x}(t_1)$ would converge to the DOA at the point $\bar{x}(t_1 + T)$. If the convergence rate is nonuniform, the point $x(t_0)$ the time $T_0$ to converge to the DOA while the point $\bar{x}(t_1)$ would take the time $T_1$ to converge to the DOA with $T_0 < T_1$. This means the point $x(t_0)$ would converge to the DOA at the point $x(t_0 + T_0)$ and the point $\bar{x}(t_1)$ would converge to the DOA at the point $\bar{x}(t_1 + T_1)$.

See Figure 2.2.2 for a graphical representation of this concept.

(a) Uniform convergence rate and uniform DOA: 
Convergence time $T(\epsilon)$ is independent of $t_0$
In many applications, it is important to know the rate at which a solution will converge to a particular equilibrium state. An inherent outcome of the DI is the convergence rate, which can be measured by the magnitude of the Lyapunov exponent. A negative value with a large absolute value indicates a fast convergence rate, likewise, a
negative value with a small absolute value indicates a slow convergence rate. Conversely, a large positive value indicates a fast divergence rate and a small positive value indicates a slow divergence rate. It is desired that the tracking error converge in the shortest amount of time and stay in an equilibrium state as long as possible. The guidance trajectory is calculated to provide the optimal path to the vehicle's final destination and any divergence from this path could cause a reduction in efficiency and timeliness in reaching the final destination. The ability for the controller to accurately follow the guidance trajectory is important so as to get the vehicle to its final destination quickly and efficiently. For the purposes of this particular application, the convergence rate is not quantitatively determined.

2.2.2 Stability

This section discusses stability in the sense of Lyapunov which includes stability, uniform stability, asymptotic stability, uniform asymptotic stability and (uniform) exponential stability of equilibria. There are numerous other definitions for stability but only stability in the sense of Lyapunov will be presented. Once again, non autonomous systems of the form \( \dot{x} = f(t, x) \) are considered. For the following definitions, consider the isolated equilibrium \( x_e = 0 \) of Equation 2.2.2

**Definition 2.2.4** The equilibrium \( x_e \) is stable if for all \( t_0 > T_0 \) and \( \epsilon > 0 \), there exists a \( \delta = \delta(\epsilon, t_0) > 0 \) such that \( \|x_0\| < \delta \) implies that \( x(t) = \|s(t, t_0, x_0)\| < \epsilon \), for all \( t \geq t_0 \); Otherwise it is said to be unstable. 

In engineering practice, this means, given a prescribed tolerance \( \epsilon \) on the behavior of \( x(t) \) from the operating point \( x_e \), an initial tolerance \( \delta = \delta(t_0) \) can be determined for \( x(t_0) \) such that the tolerance \( \epsilon \) is satisfied for all \( t \geq t_0 \). This is illustrated in Figure 2.2.3.
Definition 2.2.5 The equilibrium $x_e$ is uniformly stable if it is stable and additionally, $\delta = \delta(\epsilon)$, independent of $t_0$.

It is clear the difference between uniform stability and stability is that the initial tolerance $\delta$ is constant for all $t \geq t_0$ while stability does not demand this property. Also, the notion of nonuniform stability is the same as uniform stability except the tolerance $\delta$ is time dependent and the tolerance $\delta$ shrinks as $t \to \infty$. The expression for the tolerance is then written as $\delta(t_0 + T)$. Both uniform and nonuniform stability are illustrated in Figure 2.2.4.

None of the above forms of stability require the equilibria to be attractive, but only stay within a given tolerance. The next three forms of stability all require the equilibria to be stable as well as attractive and are all graphically depicted in Figure 2.2.4. The first definition is for asymptotic stability.

Definition 2.2.6 An isolated equilibrium $x_e$ is asymptotically stable if it is stable and attractive.
In the application of the 2GRLV nonlinear tracking control system, this means the tracking error will remain sufficiently small \( x(t) < \epsilon \) if the initial tracking error is guaranteed to be sufficiently small \( x(t_0) < \delta \). In addition, the tracking error will converge to zero as \( t \to \infty \).

**Definition 2.2.7** An isolated equilibrium \( x_e \) is uniformly asymptotically stable if it is uniformly stable and uniformly attractive.
Once again, for the 2GRLV nonlinear tracking control system, the tracking error will stay sufficiently small \( x(t) < \epsilon \) if the initial tracking error is guaranteed to be sufficiently small, but now the value of the initial tolerance \( \delta \) is time dependent. Therefore, the expression for the tracking error is \( x(t_0) < \delta(t_0 + T) \). And again, the tracking error will converge to zero as \( t \rightarrow \infty \).

**Definition 2.2.8** An isolated equilibrium \( x_e \) is (uniformly) exponentially stable if there exists constants \( \delta, k, \gamma > 0 \) such that \( \|x_0\| < \delta \) implies \( x(t) = \|s(t, t_0, x_0)\| \leq k\|x_0\|e^{-\gamma(t-t_0)} \).

Exponential stability is the strongest of the five forms of stability mentioned, as it guarantees the solution will not only satisfy the criteria for uniform asymptotic stability, but the solution will also converge to \( x_e \) exponentially as \( t \rightarrow \infty \).

All the previous definition lead to one final definition that defines the stability of a nominal trajectory. Once again, in this case, the nominal trajectory is the guidance trajectory commanded by the guidance system of the 2GRLV. See Figure 2.2.5 for a graphical depiction of this concept.

**Definition 2.2.9** Let \( \bar{x}(t) = \bar{s}(t, t_0, x_0) \) be a nominal (solution) trajectory of Equation 2.2.1, and let \( x(t) = s(t, t_0, x_0) \) be any (solution) trajectory of Equation 2.2.1 in the operating region \( \Omega \). Define the tracking error variable \( \epsilon(t) = x(t) - \bar{x}(t) \). Then the tracking error dynamics governed by

\[
\dot{\epsilon} = \dot{x} - \dot{\bar{x}} = f(t, \bar{x} + \epsilon) - f(t, \bar{x}) = g(t, \epsilon, \bar{x}(t)),
\]

has an isolated equilibrium \( \epsilon(t) \equiv 0 \). The stability of the nominal trajectory \( \bar{x}(t) \) is then defined by the stability of the null equilibrium \( \epsilon(t) \equiv 0 \).
For the presented application, the tracking controller is responsible for acquiring and stabilizing the nominal (command) trajectory. The ability for the controller to handle this task is to be determined by the DI.

2.3 Lyapunov Exponent Stability Assessment for NL Systems

In order to understand the application of Lyapunov's First Method, it is necessary to present its definition along with some discussions of its behavior. This section makes several references to the concepts of stability and equilibrium that were presented in the previous sections. It uses those concepts to explain Lyapunov's First Method (Section 2.3.1), which leads to the definition of the Lyapunov exponent for LTV systems in Section 2.3.2. Finally, the Lyapunov exponent is extended to NLT systems in Section 2.3.3.
2.3.1 Lyapunov’s First Method

The following theorem states Lyapunov’s Indirect Method for showing uniform asymptotic stability of the origin for the nonautonomous case.

**Theorem 2.3.1** Let \( x = 0 \) be an equilibrium point for the nonlinear system

\[
\dot{x} = f(t, x),
\]

where \( f : [0, \infty) \times D \rightarrow \mathbb{R}^n \) is continuously differentiable, \( D = \{ x \in \mathbb{R}^n \| x \| < r \} \), and the Jacobian matrix \( \frac{\partial f}{\partial x} \) is bounded and Lipshitz on \( D \), uniformly in \( t \).

Let

\[
A = \frac{\partial f}{\partial x}(t, x)|_{x=0}.
\]

Then, the origin is an exponentially stable equilibrium point for the nonlinear system if it is an exponentially stable equilibrium point for the linear system

\[
\dot{x} = A(t)x.
\]

\( \nabla \)

It is from this theorem that the Lyapunov exponent can be derived and is presented in the following section.

2.3.2 Definition of the Lyapunov Exponent

In this section, the definition of the Lyapunov exponent is presented. The Lyapunov exponent, denoted by \( \lambda \), may be used to measure the sensitivity of a system’s behavior to initial conditions. The notion of Lyapunov characteristic exponents is formally defined by the following.

a) Let \( x(t, t_0) \) be a (real-valued) solution to \( \dot{x} = A(t)x \) with a fixed initial time \( t_0 \). Then, the Lyapunov characteristic exponent for \( x(t, t_0) \) is defined by
The exponent provides an average rate of divergence of the trajectory from an equilibrium point which provides an indication of instability or classification of a given chaotic system for a given initial condition. If $\lambda$ is negative, trajectories close to the equilibrium point converge to the equilibrium point as $t \to \infty$ thus indicating stable or nonchaotic behavior. If, however, $\lambda$ is positive, then trajectories close to the equilibrium point diverge. This indicates the trajectory is sensitive to initial conditions and is therefore unstable or chaotic.

b) A fundamental solution matrix

$$X(t, t_0) = [x_1(t, t_0) \ x_2(t, t_0) \ \cdots \ x_n(t, t_0)]$$

is said to be normal in the sense of Lyapunov if for every fundamental solution matrix

$$Z(t, t_0) = [z_1(t, t_0) \ z_2(t, t_0) \ \cdots \ z_n(t, t_0)],$$

$$\sum_{k=1}^{n} \lambda(x_k(t, t_0)) = \nu(X) \leq \nu(Z) = \sum_{k=1}^{n} \lambda(x_k(t, t_0)).$$

The set $\{\lambda(x_k(t, t_0))\}_{k=1}^{n}$ is called a Lyapunov spectrum w.r.t. $t_0$.

c) An LTV system $\dot{x} = A(t)x$ with a fixed initial time $t_0$ is said to be regular at $t_0$ in the sense of the Lyapunov if any Lyapunov normal fundamental solution matrix $X(t, t_0)$ satisfies
\[
\nu(X) = \lim_{t \to \infty} \sup_{t_0} \frac{1}{t} \int_{t_0}^{t} \text{tr}A(\tau) d\tau.
\]

There are a few notes that should be stated about the above definition. First is that for any LTV system \( \dot{x} = A(t)x \), there exist at most \( n \) distinct Lyapunov characteristic exponents. Second, Lyapunov spectra are invariant under Lyapunov transformations

\[
B(t) = L^{-1}(t)[A(t)L(t) - \dot{L}(t)],
\]

where \( L(t), L^{-1}(t), \) and \( \dot{L}(t) \) are uniformly bounded for \( t \geq t_0 \). Finally, in general,

\[
\nu(X) \geq \lim_{t \to \infty} \sup_{t_0} \frac{1}{t} \int_{t_0}^{t} \text{tr}A(\tau) d\tau,
\]

and \( \nu(X) \) of a Lyapunov regular system attains the lower bound. Lyapunov regularity is invariant under Lyapunov transformations.

The preceding definition provides the theoretical background necessary for understanding the Lyapunov exponent and its use for instability assessment. The following theorem describes the proper behavior of the Lyapunov exponent.

**Theorem 2.3.2** A Lyapunov LTV system \( \dot{x} = A(t)x \) is asymptotically stable at \( t_0 \) with an exponential convergence rate if all Lyapunov Characteristic Exponents in the Lyapunov spectrum are negative.

\( \diamond \)

It should be noted that for LTI systems, the Lyapunov characteristic exponents coincide with the real part of the eigenvalues of the system. For Linear Periodic systems, they coincide with the real part of the Floquet characteristic exponents [18]. In these cases, the Lyapunov Theorem can be strengthened to exponential stability being equivalent to the
negativity of all Lyapunov exponents. This is also referred to as a non-chaotic system. Also, it follows from Theorem 1.3.2 that for regular LTV systems, the null solution is exponentially stable over $[T, \infty)$ if and only if the Lyapunov exponents in a Lyapunov spectrum are negative and bounded from zero uniformly for all $t_0 \geq T$.

2.3.3 Lyapunov Exponent Behavior

In this section, the behavior of the Lyapunov exponent is discussed. This discussion will first demonstrate the exact equation used in this research, including a description of its inputs and outputs. An explanation of the exponent's behavior will also be explained. First, consider the following equation to be the Lyapunov exponent for a nonlinear system.

**Definition 2.3.1** Let $x(t)$ be the scalar tracking error, $t$ be the time and $\lambda_0$ be the Lyapunov exponent

$$\lambda_0 = \lim_{t \to \infty} \sup_{t_0} \frac{\int_{t_0}^{t} \frac{\dot{x}(\tau)}{x(\tau)} d\tau}{t - t_0}. \quad 2.3.12$$

It can be shown that if $\lambda_0$ is negative, then the tracking error $x(t, t_0) \to 0$ exponentially. The fact that the negativity of $\lambda_0$ translates to a converging trajectory is further explained below.

Suppose the tracking error is positive (positive point in figure 2.3.1), in order to ensure stability, it is necessary to drive the tracking error to zero. This would represent a negative derivative, as the slope must be negative for the tracking error to reach zero. This would cause $\dot{x}(t)/x(t) < 0$ for any value of $x(t) > 0$ and $\dot{x}(t) < 0$. Conversely, if the tracking error were negative, the slope would have to be positive in order for the
tracking error to reach zero. Once again this would cause $\dot{x}(t)/x(t) < 0$ for any value of $x(t) < 0$ and $\dot{x}(t) > 0$.

Therefore, as $t \to \infty$, the value of each exponent is summed by the integral and the overall behavior should be negative to ensure stability. This method was found to produce good results for relatively "smooth" signals. However, the derivative is very sensitive to high frequency noise. This is because the derivative of a very steep slope is very large while the error is very small. This causes the numerator of the exponent term to be large and give a very sporadic behavior. This behavior along with some solutions will be discussed later.

For time-dependent systems, the system trajectory might diverge, also known as departure, momentarily so as to violate some operational constraints (i.e. excessive tracking errors), even though it may ultimately converge to zero as $t \to \infty$. In order to detect such practical instability, a short-time, or windowed, Lyapunov exponent can be defined as follows

$$\lambda_T(t) = \frac{\int_{t-T}^{t} \dot{x}(\tau) d\tau}{T},$$

where $T$ is the width of the moving window over which the exponential departure rate is observed. The value for $T$ should be chosen based on the speed of the system. If the
system varies at a rapid pace (i.e. the tracking error rapidly changes), a smaller $T$ window is necessary in order to accurately analyze the system. If the system varies at a slow pace, a larger $T$ should be used. Ultimately, it is desirable to have a time varying window size if the system varies at different frequencies through time. A suitable size of this window is experimentally determined in Chapter 3 and a more analytical method is outlined and left for future work. In the sequel, we shall call $\lambda_0$ a Long Departure Index (LDI) for the trajectory, and $\lambda_T$ a Short Departure Index (SDI). Unless otherwise noted, DI will be used to denote the short and long DI together. If only one or the other is being discussed, they will be referred as SDI and LDI respectively.
Chapter 3 Departure Index - An Instability Indicator

3.1 Introduction

This chapter discusses the concepts, principles, and implementation of the DI and all supporting algorithms. Section 3.2 is on Concepts/Principles which discusses how the DI is used in this research and what information the DI is expected to provide to other systems. In addition, this section will briefly introduce the individual supporting algorithms that assist in making the DI an effective algorithm for detecting the onset of instability.

Section 3.3, on Practical Issues and Solutions, presents the issues of implementing the LDI and SDI in all simulation packages and, in addition discusses the issues that are dealt with by the supporting algorithms. The issues will be presented, along with an explanation of why they present a problem to the function of the DI. The developed solutions to these issues will be named, but the function of these algorithms is left for explanation in the subsequent subsections.

The individual subsections titled Dead Band (Section 3.3.1), Low Pass Filter (Section 3.3.2) and Hilbert Transformation (Section 3.3.3), discuss the individual supporting algorithms meant to remedy the issues that jeopardize the DI's ability to accurately and in a timely manner indicate loss of stability. The implementation of each algorithm is explained and program code is referenced to the Appendix. The Hilbert Transformation groundwork is presented and some implementation issues are explained. The actual implementation of this algorithm is not included and left for future work.

Section 3.4 is on Implementation which discusses how the DI was implemented in SLIM, MAVERIC and IDOS. The implementations in SLIM and MAVERIC are identical and are discrete time implementations. IDOS is a continuous-time simulation environment, thus requiring a slightly different implementation. Reference will be made to the program code for all implementations located in the Appendix.
3.2 Concepts/Principles

The concept of the DI is to obtain a timely and accurate indication of the onset of instability in a dynamic NLTV system at any given state. This is desired because of the unpredictable nature of NLTV systems and the need for a technique that can indicate onset of instability and provide necessary information in order to prevent a disaster from occurring. The DI can be used with any system provided the trajectory data is readily available and a method for computation is at hand.

It is desired, for this application, to indicate the onset of instability in a timely and accurate manner. As presented in the previous chapter, the Lyapunov exponent indicates if a solution of a nonlinear differential equation diverges from an equilibrium state. Therefore, the Lyapunov exponent is being used as the DI in order to indicate the departure of the actual vehicle's trajectory from the commanded trajectory.

The DI is analyzed by an upper level system (Autocommander) that manages all IG&C (Integrated Guidance and Control) systems. The Autocommander's role is to command the proper subsystems in the event of an emergency indicated by either the DI or some other vehicle monitoring system(s). Figure 3.2.1 is a graphical depiction of the

![Figure 3.2.1 Entire Integrated Guidance and Control system](image-url)
overall IG&C system. The DI would be located in the "Attitude Control" block and would communicate any instability to the Autocommander.

It is necessary for the DI to provide not only timely indication of instability, but it must also provide accurate indication of loss of stability. Since accuracy is an important property of the DI, one must carefully examine any and all assumptions and/or approximations in the development of the algorithm. One such approximation is the calculation of a short-term Lyapunov exponent. As presented in Chapter 2, the Lyapunov exponent is defined for all time, $t \to \infty$. Since an instantaneous indication of instability is required, a short-term Lyapunov exponent is explored. This approximation introduces error into the solution which can produce inaccurate results. This inaccuracy may have an effect on the DI's ability to accurately, and in a timely manner, indicate loss of stability.

Some supporting algorithms were developed to assist in remedying some of the issues that arose when developing the DI. One such algorithm is the called the Dead Band, which prevents the possibility of a singular solution by the quantity $\dot{x}(t)/x(t)$, when $x(t) = 0$. The Dead Band avoids the possibility of a singular solution by forming a small bound about zero and either adjusts the value of the tracking error or Lyapunov exponent to prevent a singular solution from being obtained.

Another supporting algorithm is a Butterworth low pass filter. This was introduced because the DI is sensitive to persistent noise in the tracking error which is characterized by high frequency noise. If the tracking error signal consistently oscillated about zero, the exponent would produce a zero result and the algorithm would not prove to be effective.

A Hilbert Transformation has been explored to remedy the singular solution issue, since the Dead Band introduces error into the solution of the DI. It is believed that the HT will provide a solution with less error, provided the tracking error is an analytic signal. If the signal is not analytic, a band pass filter can be introduced to resolve the problem.
addition, the DI only calculates the Lyapunov exponent of the dominant (slowest) mode. An algorithm called the Hilbert-Huang Transformation, identifies all modes and applies the HT for each mode. This would allow the Lyapunov exponent to be calculated for other modes that might become excited under perturbations, thus making the DI a more effective indicator of instability.

3.3 Practical Issues and Solutions

There are some important issues that needed to be addressed while developing the DI. Among these issues are that of persistent noise, that is either due to external or internal sources, the possibility of obtaining a singular solution and the fact that the DI is only calculated for the slowest (dominant) mode. These issues presented a significant challenge to the implementation of the DI and it was necessary to explore some possible solutions. This section will discuss the issues and how they affected the DI.

The first issue is that of the persistent noise in the tracking error signal. High frequency noise presents a problem because the tracking error evaluated by the DI can oscillate about zero, causing the trajectory to never converge and the end result is a zero exponent. This is because the constant positive and negative values of the DI eventually cancel each other. The other problem with the noise is that the Lyapunov exponent is no longer that of the actual trajectory, but that of the trajectory produced by the system along with the persistent disturbance and noise. The figure below is a block diagram of the SDI integrated with the controller, the NLTV plant, sensors and the additional disturbances.

The disturbances are those elements labeled "Persistent Disturbance" and "Wide band Noise" and are not always directly measurable or easily removed. In many cases, they cause highly undesirable effects on the system's performance and must be detected and dealt with in the proper fashion. The DI can help in identifying if an external
disturbance is present, but it is unable to indicate the exact nature of the disturbance. Figure 3.3.1 shows the location of the disturbances to the system.

Figure 3.3.1 SDI block diagram with disturbances

The simulation results shown in Figure 3.3.2 (a and b) demonstrate the issue of the persistent noise. The plot in (a) is a full view of the LDI (blue), SDI (green) and the tracking error (red) for the inner-loop yaw channel of the TLC controller. The noise is more prominent in the inner-loop as the bandwidth of the inner-loop is much higher than the outer-loop. Figure 3.3.2 (b) shows a blown up view of the plot and the high frequency noise in the tracking error (red) is clearly seen. By adding the low pass filter and the Dead Band, the issue of the persistent noise is remedied, which can be seen in Figure 3.3.3. Figure 3.3.3 is a plot of the outer-loop LDI (blue), SDI (green) and the tracking error (red). The expected trend of the SDI is clearly seen. As the tracking error diverges from zero in either the positive or negative direction, the SDI becomes more positive. On the other hand, as the tracking error converges to zero, the SDI becomes increasingly negative.
Figure 3.3.2 Inner-loop yaw LDI, SDI and tracking error before low pass filter and Dead Band introduced.
Another major issue is the possibility of the existence of a singular solution. If the tracking error \( x(t) \) is identically zero, the integrand \( \dot{x}(t)/x(t) \) produces a singular solution. In this case, it is necessary for an approximation to be made to prevent this problem from occurring, thus introducing more uncertainty in the accuracy of the solution.

In a NLTV system as complex as the RLV, it is not uncommon that numerous operational modes are present in the system. A limitation of the DI is that it is only able to calculate the Lyapunov exponent for the slowest mode. While this mode usually poses the most obvious threat to the stability of the system's motion, the underlying modes might become excited under certain perturbations and cause the system's motion to become unstable. Therefore, the identification of these "hidden" modes might allow the calculation of a Lyapunov exponent for each mode. This would make the DI a more accurate indicator of loss of stability and increase the number of instability classifications. 

Figure 3.3.3 Outer-loop yaw LDI, SDI and tracking error after low pass filter and Dead Band introduced
if one of the hidden modes were to become excited and cause the trajectory to go unstable. A possible solution to this issue is presented in Section 3.3.3.

The presented issues pose significant problems to the implementation of the DI. The following sections aim to provide solutions to these problems by filtering the noise, or using specific methods to approximate the solutions of the DI. Subsections 3.3.1 and 3.3.2 present solutions that have been implemented and tested, while Section 3.3.3 simply presents the foundation for a future solution.

3.3.1 Dead Band

The Dead Band is used to prevent a singular solution from occurring and also masks some of the small noise. The idea of the Dead Band is that a small value \( \pm \varepsilon \) is chosen to form a region about zero (tracking error) and following the logic explained below, approximates the tracking error or the Lyapunov exponent. The figure below illustrates this logic and a description is provided.

![Figure 3.3.4 Graphical representation of Dead Band logic](image)

The logic of the Dead Band is as follows.

The list's index numbers correspond to those in figure 3.3.4.

1) Any positive tracking error \( x(t) \) that has a previous value outside the bound \( \varepsilon \) and current value inside the bound \( \varepsilon \), the tracking error \( x(t) \) is set to \( \varepsilon \).
2) Any negative tracking error $x(t)$ that has a previous value outside the bound $-\epsilon$ and current value inside the bound $-\epsilon$, the tracking error $x(t)$ is set to $-\epsilon$.

3) Any positive tracking error $x(t)$ that has a previous value inside the bound $\epsilon$ and current value outside the bound $\epsilon$, the tracking error $x(t)$ is left alone.

4) Any negative tracking error $x(t)$ that has a previous value inside the bound $-\epsilon$ and current value outside the bound $-\epsilon$, the tracking error $x(t)$ is left alone.

5) Any positive tracking error $x(t)$ that has a previous and current value inside the bound $\epsilon$, the exponent $\lambda$ is set to zero.

6) Any negative tracking error $x(t)$ that has a previous and current value inside the bound $-\epsilon$, the exponent $\lambda$ is set to zero.

This implementation eliminates the possibility of a singular solution and assists in reducing the small noise, but introduces error into the solution. In some cases, this error might cause serious problems. However, for this application, the error has not been determined to have an unacceptable effect. Different values of $\epsilon$ can be chosen to produce different results. The larger the $\epsilon$, the more error that can be introduced into the solution. An acceptable value of $\epsilon$ has been chosen as $\epsilon=.001$.

### 3.3.2 Low Pass Filter

A low pass filter is used to smooth the tracking error before being evaluated by the DI. This removes the high frequency noise that produces undesirable results. For the implementation in MAVERIC/SLIM, a discrete time implementation of a Butterworth filter was used. The coefficients for the filter's equations were calculated using the "Butter" command in MATLAB, which requires the order of the desired filter and the
desired cutoff frequency as inputs. A second order filter was designed and the cutoff frequency was chosen to eliminate the effects of the 1 Hz harmonic produced by the guidance command. The resulting coefficients give the following equation for the filter:

$$H(z) = \frac{0.097768z^2 - 0.19403z + 0.097768}{1.0z^2 - 1.9471z + 0.94857}.$$ 3.3.1

The program code for this algorithm is shown in Appendix C.

For the implementation in SIMULINK, a continuous time filter was needed because of the continuous nature of the simulation. A continuous time implementation of the Butterworth filter could have been used, but the Pseudo-Differentiator used to differentiate the tracking error, produces a low pass filtered tracking error. The Pseudo-Differentiator is shown in Appendix B.

Once again, filtering the noise introduces error into the system, however, if the noise were not removed, the exponent would produce a zero exponent, thus never producing the correct results. The error introduced by the filter can be considered negligible as the noise has a small effect on the performance of the system in general.

### 3.3.3 Hilbert Transformation

The use of the Dead Band to ensure a singular solution is avoided is a good placeholder for now, however, this method introduces error into the calculation of the DI. This error can have varying effects depending on the speed at which the system varies. If the system rapidly changes, this error could be a significant problem. On the other hand, if the system varies slowly, this error might be negligible. Nonetheless, a more accurate method is desired and such a method is the Hilbert Transformation. There are a few drawbacks to this method and are presented in this section. One such issue is the need for the tracking error to be an analytic signal. The assumption can be made that this is so, or a band pass filter can be introduced in order to assure the signal is an analytic one.
The Hilbert Transformation approximates the imaginary solution \( y(t) \) from a known real solution \( x(t) \) and vice versa. The Hilbert Transformation is defined for a real and imaginary approximation by the equations below [3]

\[
x(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{y(\tau)}{t - \tau} \, d\tau,
\]

\[
y(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} \, d\tau.
\]

Since the known signal is the real valued part, we are only concerned with Equation 3.3.3. One should notice that the equation calls for all time \(- \infty < t < \infty\), which for practical applications is not possible. Therefore, we only consider the portion from \( t_0 \rightarrow t \) thus giving the following derivation:

\[
y(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} \, d\tau
\]

\[
= \frac{1}{\pi} \left[ \int_{-\infty}^{t_0} \frac{x(\tau)}{t - \tau} \, d\tau + \int_{t_0}^{t} \frac{x(\tau)}{t - \tau} \, d\tau + \int_{t}^{\infty} \frac{x(\tau)}{t - \tau} \, d\tau \right]
\]

\[
= \frac{1}{\pi} \left[ 0 + \int_{t_0}^{t} \frac{x(\tau)}{t - \tau} \, d\tau + 0 \right]
\]

\[
= \frac{1}{\pi} \int_{t_0}^{t} \frac{x(\tau)}{t - \tau} \, d\tau.
\]

Now that a final expression for the transformation is achieved, an explanation of its implementation is in order. The real solution \( x(\tau) \) is the tracking error of the controller. In this case, \( \tau \) is the time step variable for the simulation. In other words, it represents each time step from initial time \( t_0 \) to the current time \( t \). This means that all values of the tracking error must be retained in order to calculate the imaginary solution. This presents a significant problem, as the simulation time of the system in question is quite long (about 3.5 minutes) and the time step is rather small (50 steps a second). This is an unacceptable amount of data to retain, so a short-term approximation is necessary.
The consequences of such an approximation have not been quantified and are left for future work.

Once the imaginary solution is obtained, the new tracking error can be expressed in the form of a complex solution \( a(t) = x(t) + iy(t) \), where \( i \) is, of course, the standard imaginary number \( \sqrt{-1} \). The calculation of the new SDI can then be expressed as follows.

\[
\lambda(t - T, t) = \frac{\int_{t-T}^{t} \frac{\dot{a}(\tau)}{a(\tau)} d\tau}{T}
\]

where

\[
\frac{\dot{a}(t)}{a(t)} = \frac{\dot{x}(t) + i\dot{y}(t)}{x(t) + iy(t)}
\]

\[
= \frac{\dot{x}(t) + i\dot{y}(t)}{x(t) + iy(t)} \cdot \frac{x(t) - iy(t)}{x(t) - iy(t)}
\]

\[
= \frac{\dot{x}(t)x(t) + i\dot{x}(t)y(t) - i\dot{y}(t)x(t) + \dot{y}(t)y(t)}{x(t)^2 + y(t)^2}
\]

\[
= \frac{\dot{x}(t)x(t) + i\dot{x}(t)y(t) + x(t)\dot{y}(t)}{x(t)^2 + y(t)^2},
\]

which leads to the complex expression

\[
\frac{\dot{a}(t)}{a(t)} = \frac{\dot{x}(t)x(t) - \dot{y}(t)y(t)}{x(t)^2 + y(t)^2} + \frac{i(\dot{x}(t)y(t) + x(t)\dot{y}(t))}{x(t)^2 + y(t)^2}.
\]

Now ignoring the imaginary term, we are left with

\[
\frac{\dot{a}(t)}{a(t)} = \frac{\dot{x}(t)x(t) - \dot{y}(t)y(t)}{x(t)^2 + y(t)^2},
\]

which then results in the final expression for the SDI

\[
\lambda(t - T, t) = \frac{\int_{t-T}^{t} \frac{\dot{x}(\tau)x(\tau) - \dot{y}(\tau)y(\tau)}{x(\tau)^2 + y(\tau)^2} d\tau}{T}.
\]
Since the tracking error is an analytic signal, the quantity $x(t)^2 + y(t)^2$ will never result in a zero solution. Thus guaranteeing to never have a singular solution for the Lyapunov exponent.

As mentioned in Section 3.3, the DI only calculates the Lyapunov exponent for the slowest mode of the system, leaving the possibility of a missed detection of instability due to a faster mode. A possible solution to this problem can be derived from the Hilbert-Huang Transformation. The Hilbert Transformation is being explored to calculate the imaginary part of the signal from the real part of the complex valued tracking signal. Huang [8] improves the HT by identifying all the modes present in a NLTV system by use of an Empirical Mode Decomposition. For the RLV, this would be done using a bank of low pass filters to identify the individual modes. The HT would calculate the imaginary solution for each mode, and the DI would calculate a Lyapunov exponent for each mode. There is still much work to be done in order to successfully implement the Hilbert Transformation or the Hilbert-Huang Transformation, but the basic problems have been identified and success is probable.

3.4 Implementation

The DI has been implemented in three simulation packages. The first two packages are MAVERIC and SLIM, which are both discrete time simulation packages. The other implementation is in IDOS, which has a continuous-time simulation environment. This section will present all implementations.

3.4.1 Discrete Time

The version of the DI in MAVERIC and SLIM is written in the C language, and is a discrete-time implementation of the algorithm. The implementations are identical, so the following explanation applies to both. The algorithm takes the tracking error and the
tracking error rate of the nonlinear tracking controller as the input and outputs the Lyapunov exponent. Figure 3.4.1 is a graphical depiction of the setup.

![Figure 3.4.1 SDI block diagram with other subsystems](image)

Note the DI has no effect on the tracking error seen by the rest of the system. It is simply used as a monitor and any algorithms applied to $x(t)$ before it is evaluated by the DI have no effect on the rest of the IG&C system.

The Euler integration method was used to calculate the integral of the quantity $\dot{x}(t)/x(t)$. This method is shown below, where the current integral is denoted by $y$, $k$ is the current time step and $x$ is the integrand $\dot{x}(t)/x(t)$:

$$y[k] = y[k - 1] + \frac{[x[k] - x[k - 1]]}{x[k]}.$$  \hspace{1cm} 3.4.1

Furthermore, the SDI was implemented using a sliding window of length $T$. Thus, the quantity $y[k - 1]$ in Equation 3.4.1 represents all past integral values back to the time $t - T$. The values of $y[k - 1]$ were shifted in much the same manner as a typical shift register. The derivative of the tracking error $x(t)$ was calculated by subtracting the previous tracking error from the current tracking error as seen by the quantity $[x[k] - x[k - 1]]$ in Equation 3.4.1. Figure 3.4.2 is a plot demonstrating the algorithm in
MAVERIC. The expected trend of a converging tracking error (red) producing a negative SDI (green) and vice versa is seen. Those areas where larger tracking errors occur, correspond to those areas where the SDI is most positive.

Figure 3.4.2 LDI, SDI and tracking error for outer-loop roll channel

3.4.2 Continuous Time

The implementation in IDOS is slightly different from the version in MAVERIC because MAVERIC is a discrete-time simulation and IDOS is a continuous time simulation. The block diagram of the SDI implementation in IDOS is shown Figure 3.4.3.

Figure 3.4.3 Block diagram of SDI implementation in IDOS
A Pseudo-Differentiator is used to calculate the derivative of the tracking error and acts as a low pass filter which is different from the Butterworth filter used in MAVERIC/SLIM. For now, the Short DI block will be shown in Figure 3.4.4 to help illustrate its implementation. In the figure below, the transport delay is the sliding window $T$ described in Section 2.3.3. For this implementation, the window is set to 5 seconds.

![Figure 3.4.4 Block diagram of Short DI integral](image)

The transport delay holds all tracking error data for a specified time range (5 seconds) and only outputs the tracking error data until the delay time is over. Until then, zero is the input to Integrator1. The simulation was run with a variable step size and the maximum and minimum step sizes were set to "auto". The integration method is the ode45 (Dormand-Prince) method. For a view of all the individual components seen in figure 3.4.3, see Appendix B. A plot of the SDI and tracking error for the outer-loop pitch channel from IDOS is shown in Figure 3.4.5. Note the LDI was not implemented in IDOS as it did not prove effective in MAVERIC/SLIM.

Notice the expected response of the SDI is present in Figure 3.4.5. The divergence of the tracking error causes the SDI to become more positive and a convergence of the tracking error causes the SDI to become more negative. Towards the end of the
simulation, the tracking error oscillates about a zero tracking error and the SDI maintains zero value due to the Dead Band. The oscillations in the tracking error made it necessary to introduce a low pass filter to avoid a zero exponent. This issue is discussed in the next section.

Figure 3.4.5 Outer-loop pitch channel SDI and tracking error in IDOS
Chapter 4 Results

4.1 Introduction

In order to demonstrate the effectiveness of the DI, extend the notion of gain/phase margins to NLTV systems and to determine an appropriate value for the short-time window $T$, some test scenarios were developed. This involved three tests, the purpose of the first test was to demonstrate the accuracy of the DI, the second test was designed to extend the use of gain/phase margins as an indication of controller robustness to NLTV systems, and the third test was used to experimentally determine the short-time window $T$. The first two tests (Sections 4.2 and 4.3) were each simulated in SLIM version 1.2 during the Ascent and Entry phases with different gain and delay margins for the controllers and analyzed for DI accuracy. The results showed a close correlation between gain and delay margin and controller effectiveness and also showed good performance by the SDI.

The third test (Section 4.4) was simulated in MAVERIC version 7.6.5 and used the controller parameters of Case 4 of the first two tests. Three different $T$ windows were used (50, 250, 500 and 1000 simulation steps = 1, 5, 10, 20 seconds respectively) and 200 dispersion simulations for each window were conducted and analyzed during the Ascent phase. Entry was not analyzed as this test was meant to just present a baseline for future analytic determination of a proper window of $T$. Some data is not included in this thesis as it is ITAR (International Traffic and Arms Regulations) restricted. Statistics on the results are provided, along with references to the documents that publish this data.

4.2 DI Accuracy and Promptness

The purpose of this test was to demonstrate the effectiveness (success rate and false alarm rate) of the DI for instability detection. The DI algorithm should correctly identify the onset of instability by showing a positive value in a timely manner, and it
should not give a false alarm when the tracking error is stable. Early detection of the onset of instability is crucial for the Autocommander to provide early warning to the necessary subsystem(s) in order to correct the instability, or explore the possibility of issuing some type of an abort. Successful recovery of stability or intact abort will save the vehicle, and, in the event of imminent danger to crew, early warning will enable safe crew bailout.

The DI algorithm was implemented in the Ascent and Entry controllers in SLIM. Both the LDI and SDI (with a most effective window size to be determined in Test 3; a 5 second window was used for this test) were analyzed. The DI data was compared to the tracking error data for each of the test cases specified in Test 2.

The Trajectory Linearization Controller (TLC) was implemented in SLIM with fixed controller gains for five different sets of gain values as specified in Table 4.2.1 below.

<table>
<thead>
<tr>
<th>Case</th>
<th>Wn roll</th>
<th>Wn pitch</th>
<th>Zeta roll</th>
<th>Zeta pitch</th>
<th>Zeta yaw</th>
<th>Gain Margin</th>
<th>Delay Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>0.14</td>
<td>0.14</td>
<td>0.14</td>
<td>0.5</td>
<td>0.5</td>
<td>1.5</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.7</td>
<td>0.7</td>
<td>1.5</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.9</td>
<td>0.9</td>
<td>1.5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>1.5</td>
<td>1.5</td>
<td>1.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes: Wn is the natural frequency and Zeta is the damping coefficient of the closed-loop eigenvalue of each channel in the outer loop. The inner loop controller gains remain the same for each case.

The controller gains were selected to result in a different level of robustness for Test 2. For each design case, 100 dispersion simulations in SLIM were performed. In the sequel, each dispersion run will be denoted by nm, where n is the case number, and mm is the run number, e.g. 2-78 denotes Run 78 in Case 2.

The large number of unstable runs due to the poor stability margin in these test cases constitutes a good test bed for the effectiveness of the DI for detecting onset of
instability. For the SDI in Tests 1 and 2, the quantity $\dot{x}(t)/x(t)$ was integrated over a 5-second time interval ($T = 250$ simulation steps), and the LDI was integrated from the initial time until the current time.

For all five test cases, the tracking error tolerances were specified as 20 degrees for Euler roll angle, and 10 degrees for Euler pitch and yaw angles, and 10 degrees/second for body roll rate, pitch rate and yaw rate. These values can be seen in Table 4.2.2

Table 4.2.2 Error threshold values for inner and outer loop ascent control

<table>
<thead>
<tr>
<th>Maximum Tracking Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

A test run is considered successful if all tracking errors are within these bounds. For each successful run, the maximum SDI and LDI in each channel was documented in Tables 2 (a)-(e) in [19]. The maximum of all maximum SDIs and LDIs in each channel was rounded up to the second digits after the decimal point and used as the DI threshold for departure detection, which are shown in Table 4.2.3. Namely, departure, or onset of instability, is declared if the DI threshold is exceeded in any channel.

Table 4.2.3 SDI and LDI thresholds for each case

<table>
<thead>
<tr>
<th>Cases</th>
<th>Short-Time DI Threshold</th>
<th>Long-Time DI Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Roll</td>
<td>Pitch</td>
</tr>
<tr>
<td>Case 1</td>
<td>0.97</td>
<td>0.86</td>
</tr>
<tr>
<td>Case 2</td>
<td>1.00</td>
<td>0.83</td>
</tr>
<tr>
<td>Case 3</td>
<td>0.87</td>
<td>0.81</td>
</tr>
<tr>
<td>Case 4</td>
<td>1.11</td>
<td>0.84</td>
</tr>
<tr>
<td>Case 5</td>
<td>0.64</td>
<td>0.60</td>
</tr>
</tbody>
</table>
It is noted that there are two distinct scenarios in the failed runs. First, the failure is caused by instability, or departure. Second, the failure is due to momentary and minor violation of the tracking error tolerance, but nonetheless the system's motion remains stable. Thus, it is desired that the DI is able to distinguish these two scenarios. If the DI exceeds its threshold in the second scenario, it is considered a false alarm.

In order to determine the effectiveness of SDI and LDI in detecting departures while avoiding false alarms, for each failed run in all five test cases, the time when the tracking error, SDI and LDI first exceeded the respective threshold were recorded in Tables 4 (a)-(e) in [19]. From these tables, it is clear the LDIs are not effective at detecting departures. However, the SDI detected departure in all failure runs except for Run Numbers 1-77, 1-90, 2-90, 3-90, 4-40, 4-79, where one tracking error exceeded the tolerance at some time, but none of the SDIs exceeded the threshold.

In order to determine whether the SDIs in Run Number 1-77, 1-90, 2-90, 3-90, 4-40, 4-79 missed detection of a departure or avoided a false alarm, and to ensure that there was no false alarm in the other failed runs, the maximum tracking error and SDI in each channel for failed runs were documented in Tables 5 (a)-(e) in [19]. The data shows that 6 runs identified indeed correctly avoided false alarms, and there were no other false alarms.

Based on the above observations, the SDI can correctly classify the successful runs and temporary tracking error violation from true departures in all 500 runs. However, a close examination of data reveals that in Run Numbers 1-91 (roll rate), 4-3 (roll rate), 4-39 (pitch rate), 5-6 (roll rate), 5-17 (roll rate), 5-59 (pitch rate), 5-71 (roll rate), 5-100 (yaw), the SDI in the indicated channels missed the detection of a departure. This is because the positive SDI threshold, while necessary for avoiding false alarms due to the noise in the tracking error signal and the short-time nature of the SDI, may shield a
departure with a positive Lyapunov exponent less than the threshold. One possible solution is to use multiple short-time window sizes which is explored in Test 3.

4.3 Delay and Gain Margin Tests

The purpose of this test was to demonstrate the effectiveness of the gain margin and delay margin as robustness indicators. A higher gain margin would enable the control system to tolerate larger parametric modeling errors, parameter variations, adverse flight conditions and vehicle performance deteriorations and faults. A higher delay margin would allow the control system to accommodate dynamic modeling errors, singular perturbations, and parasitic dynamic modes such as spin departure, rocking and flutter. A robust guidance and flight control system would significantly reduce the need for abort, and increase the success rate of intact abort or safe crew bailout when necessary.

The DI algorithm and a variable gain delayer were implemented in a trajectory linearization controller (TLC) in SLIM. The controller gain values were tuned to arbitrary values about the nominal design and the gain and delay margins were determined and recorded using the proposed algorithm. Then, 100 dispersion simulations, with identical settings for each tuning case, were run and the failure ratio, i.e. the number of failed runs vs. the total (100) runs, were recorded. A sufficiently high positive correlation between the high gain and delay margin of a tuning case and low failure ratio shall be considered as a success of the proposed gain and delay margin determination algorithm.

In accordance with the testing plan, the TLC controller with 5 different control gain designs were implemented as shown in Table 4.2.1 above. Then 100 Monte Carlo dispersion runs were performed as described above for Test 1. The gain margins and delay margins were determined experimentally and shown in the right most columns in Table 4.2.1. Table 4.3.1 shows all failed runs, where the failure is defined as one or more tracking errors exceed the tolerance bounds shown in Table 4.2.2.
In addition, Table 4.3.2 shows all runs that were terminated by a "core dump" due to excessive tracking error departures. A "core dump" is when the simulation terminates due to a mathematical error which is indicative of excessive tracking errors. Simulations that end in core dumps are worse than those that simply end in failure. The positive correlation between the high gain and delay margin of a tuning case and low failure ratio is evident, thus proving the concept of using the gain margin and delay margin as a
stability robustness metric. Further study is needed to develop an analytical method for determining the gain and delay margins for NLTV systems.

4.4 Short DI Window Size Tests

The purpose of this test is to experimentally determine the proper size of the SDI window \( T \). The original number was chosen to be 250 simulation steps, which resulted in a 5 second window. This number was chosen rather randomly and was never changed as it produced good results. For a system with rapidly changing dynamics, a short-time window is desired as a window that is too long will produce results with poor resolution. This would cause the DI to operate incorrectly, most likely missing instability detection or declaring instability too late. Conversely, if the system dynamics are slow and a smaller time window is chosen, the SDI might declare instability when the system's motion is actually stable.

The tests were conducted using the TLC controller in MAVERIC version 7.6.5. Four different values of the time window were chosen (50, 250, 500 and 1000 simulation steps = 1, 5, 10, 20 seconds respectively) and each case was simulated, during the Ascent phase, for 200 dispersions. The bandwidth and damping ratios for Case 4 from Tests 1 and 2 were used for this test to ensure numerous failures. See Table 4.2.1 for these values. It should be noted that each of the four cases was run with the exact same 200 dispersions. This means, run number 1 for the first case is exactly the same as cases 2, 3 and 4. This was done to ensure each case could be legitimately compared to every other case. After the tests were conducted, the failure runs were analyzed as presented in [19]. Specifically, the SDI failure times were subtracted form the tracking error times for each channel. If a detection was not made by both the tracking errors and SDIs, the number 9999.99 is displayed. If one or the other detects instability and the other doesn't, the result is the subtraction of the time of indication from the number 9999.99. If the number is
positive, the SDI indicated instability while the tracking error did not and if the number is negative, the tracking error made the detection. If both methods detect instability and the subtraction results in a positive number, the SDI detected the failure first and the number of times this occurs is tallied. Otherwise, the tracking error detected the failure first. These statistics were separated into two tables, one for the outer-loop and the other for the inner-loop. This is because the inner-loop is much faster than the outer-loop and the best values for $T$ differ between loops. The statistics are shown for all four cases in Table 4.4.1.

Table 4.4.1 Summary for number of SDI detections prior to tracking error detections

(a) Outer Loop

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Roll</th>
<th>Pitch</th>
<th>Yaw</th>
<th>Total Number of Detections</th>
<th>Total Number of Unique Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 50$</td>
<td>28</td>
<td>0</td>
<td>1</td>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>$T = 250$</td>
<td>39</td>
<td>1</td>
<td>0</td>
<td>40</td>
<td>39</td>
</tr>
<tr>
<td>$T = 500$</td>
<td>27</td>
<td>1</td>
<td>0</td>
<td>28</td>
<td>27</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>24</td>
<td>1</td>
<td>0</td>
<td>25</td>
<td>24</td>
</tr>
</tbody>
</table>

(b) Inner Loop

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Roll Rate</th>
<th>Pitch Rate</th>
<th>Yaw Rate</th>
<th>Total Number of Detections</th>
<th>Total Number of Unique Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 50$</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>$T = 250$</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$T = 500$</td>
<td>4</td>
<td>0</td>
<td>10</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>$T = 1000$</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>15</td>
<td>14</td>
</tr>
</tbody>
</table>

The column titled "Total Number of Detections" is the sum of the numbers in the first six columns. The last column counts the number of the total number of simulations classified as failures, not counting multiplicities. This number is more meaningful than the previous column as detection in only one channel is required to indicate instability.
There were a total of 91 simulation runs that ended in a failure according to the SDI and tracking errors. As can be seen in the previous table, the SDI indicated instability before the tracking error, only 45% of the time for $T = 250$. However, further study of the results, indicate a number of false alarms occurred due to the tracking error in the inner-loop pitch channel. On runs 1, 3, 5, 6, 41, 63, 64, 73, 76, 77, 82, 98, 111, 126, 146, 153, 162, 166, 187 the tracking error threshold was exceeded, while the SDI threshold had only two false alarms. The SDI thresholds are shown in the table below.

<table>
<thead>
<tr>
<th>SDI Thresholds for Each T Value</th>
<th>Roll</th>
<th>Pitch</th>
<th>Yaw</th>
<th>Roll Rate</th>
<th>Pitch Rate</th>
<th>Yaw Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Roll</td>
<td>Pitch</td>
<td>Yaw</td>
<td>Roll Rate</td>
<td>Pitch Rate</td>
<td>Yaw Rate</td>
</tr>
<tr>
<td>50</td>
<td>1.1</td>
<td>1.3</td>
<td>1.9</td>
<td>1.5</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>250</td>
<td>1.1</td>
<td>1.3</td>
<td>1.9</td>
<td>1.5</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>500</td>
<td>0.7</td>
<td>0.7</td>
<td>1</td>
<td>0.7</td>
<td>0.9</td>
<td>0.5</td>
</tr>
<tr>
<td>1000</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
</tr>
</tbody>
</table>

In Figure 4.4.1 (a and b), a false alarm case is plotted. Figure 4.4.1 (a) is the inner-loop tracking errors with the green line representing the pitch rate channel, which clearly exceeds the 10 degree threshold and the simulation does not go unstable. This is the channel that caused the false alarm. In Figure 4.4.1 (b) is the SDI for the inner-loop, where the green line is the pitch rate. It is seen that the SDI does not exceed the threshold of 1.7 as documented in Table 4.4.2.
(a) Inner-Loop tracking errors

(b) Inner-Loop SDI
(c) Inner-loop pitch rate LDI, SDI and Tracking Error

Figure 4.4.1: Inner-Loop tracking errors and SDIs for a false alarm run

These 19 false alarms reduce the number of overall failures to 72, thus giving the SDI a 55% rate of indication for instability before the tracking error. Furthermore, the SDI only had two false alarms and the time difference between an indication by tracking error versus that of the SDI is rather insignificant. In most cases, the tracking error only beat the SDI by less than 10 seconds. In addition, critical instability usually happens a number of seconds after an alarm is thrown, which should leave enough time to try and correct the manner or abort the mission.

By studying these results for all four different values of $T$, it is concluded that the 5 second window is the best choice for the outer-loop and the 1 second window is the best choice for the inner-loop. All other cases resulted in more false alarms as indicated in Table 4.4.3. While both the 250 and 500 step window sizes produced the same number of false alarms, the statistics displayed in Table 3.4.1 show the 250 step size detected instability before the tracking error 39 times as opposed to the 28 times for the 500 step
size. This can be attributed to the fact the window was too long, therefore the system
dynamics were varying at a frequency too high to be detected by this size window. In
addition, the other window sizes had more false alarms and slower response times. It
should be noted that the SDI indicated false alarms on the same simulation runs as the
tracking error.

Table 4.4.3 Number of false alarms by the SDI

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Run Numbers of SDI False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>6, 76, 146, 153, 162</td>
</tr>
<tr>
<td>250</td>
<td>146, 187</td>
</tr>
<tr>
<td>500</td>
<td>146, 187</td>
</tr>
<tr>
<td>1000</td>
<td>3, 6, 76, 146, 166, 187</td>
</tr>
</tbody>
</table>

The maximum values of the SDIs (SDI thresholds) were tallied as well as the
maximum values of the minimum SDIs for the failed runs. That is, for the simulation that
produced the smallest SDI maximum value and ended in a failure, the maximum value of
that run was tallied. This was executed for all 4 different values of $T$. The numbers for
the corresponding channels were subtracted from each other. This gives a SDI region of
detection and gives some insight into the choice of the SDI threshold. If the subtraction
ended in a negative number, there was a region for which the SDI was unable to indicate
instability. Thus resulting in a misclassification. If, however, the number is positive,
instability should have been detected. These statistics are shown in Tables 4.4.4 (a-d).

The results in Table 4.4.4 show that in most cases, the threshold for the SDI was
chosen in such a way that left the possibility for a missed detection of instability. Every
case where a negative number is displayed, a detection could have been made sooner or
avoided a missed detection if the threshold were chosen more appropriately.
Table 4.4.4 SDI threshold statistics

(a) Threshold for $T = 50$

<table>
<thead>
<tr>
<th>Roll Pitch Yaw Roll Rate Pitch Rate Yaw Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Success</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.3</td>
</tr>
<tr>
<td>1.9</td>
</tr>
<tr>
<td>1.5</td>
</tr>
<tr>
<td>1.7</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>Min-Max Failed</td>
</tr>
<tr>
<td>4.688</td>
</tr>
<tr>
<td>6.534</td>
</tr>
<tr>
<td>4.276</td>
</tr>
<tr>
<td>2.796</td>
</tr>
<tr>
<td>4.485</td>
</tr>
<tr>
<td>4.224</td>
</tr>
<tr>
<td>Min-Max</td>
</tr>
<tr>
<td>3.588</td>
</tr>
<tr>
<td>5.234</td>
</tr>
<tr>
<td>2.376</td>
</tr>
<tr>
<td>1.296</td>
</tr>
<tr>
<td>2.785</td>
</tr>
<tr>
<td>3.124</td>
</tr>
</tbody>
</table>

(b) Threshold for $T = 250$

<table>
<thead>
<tr>
<th>Roll Pitch Yaw Roll Rate Pitch Rate Yaw Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Success</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.3</td>
</tr>
<tr>
<td>1.9</td>
</tr>
<tr>
<td>1.5</td>
</tr>
<tr>
<td>1.7</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>Min-Max Failed</td>
</tr>
<tr>
<td>1.047</td>
</tr>
<tr>
<td>1.582</td>
</tr>
<tr>
<td>0.65</td>
</tr>
<tr>
<td>0.79</td>
</tr>
<tr>
<td>1.061</td>
</tr>
<tr>
<td>0.896</td>
</tr>
<tr>
<td>Min-Max</td>
</tr>
<tr>
<td>-0.053</td>
</tr>
<tr>
<td>0.282</td>
</tr>
<tr>
<td>-1.05</td>
</tr>
<tr>
<td>-0.71</td>
</tr>
<tr>
<td>-0.639</td>
</tr>
<tr>
<td>-0.204</td>
</tr>
</tbody>
</table>

(c) Threshold for $T = 500$

<table>
<thead>
<tr>
<th>Roll Pitch Yaw Roll Rate Pitch Rate Yaw Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Success</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0.7</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>Min-Max Failed</td>
</tr>
<tr>
<td>0.616</td>
</tr>
<tr>
<td>0.833</td>
</tr>
<tr>
<td>0.476</td>
</tr>
<tr>
<td>0.439</td>
</tr>
<tr>
<td>0.585</td>
</tr>
<tr>
<td>0.462</td>
</tr>
<tr>
<td>Min-Max</td>
</tr>
<tr>
<td>-0.084</td>
</tr>
<tr>
<td>0.133</td>
</tr>
<tr>
<td>-0.522</td>
</tr>
<tr>
<td>-0.261</td>
</tr>
<tr>
<td>-0.315</td>
</tr>
<tr>
<td>-0.038</td>
</tr>
</tbody>
</table>

(d) Threshold for $T = 1000$

<table>
<thead>
<tr>
<th>Roll Pitch Yaw Roll Rate Pitch Rate Yaw Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Success</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>Min-Max Failed</td>
</tr>
<tr>
<td>0.194</td>
</tr>
<tr>
<td>0.217</td>
</tr>
<tr>
<td>0.176</td>
</tr>
<tr>
<td>0.137</td>
</tr>
<tr>
<td>0.24</td>
</tr>
<tr>
<td>0.228</td>
</tr>
<tr>
<td>Min-Max</td>
</tr>
<tr>
<td>-0.206</td>
</tr>
<tr>
<td>-0.183</td>
</tr>
<tr>
<td>-0.325</td>
</tr>
<tr>
<td>-0.183</td>
</tr>
<tr>
<td>-0.16</td>
</tr>
<tr>
<td>-0.072</td>
</tr>
</tbody>
</table>

Until an analytic method for determining the proper window size is developed, 5 and 1 seconds, for the outer and inner loops respectively, appear to be good choices. Ultimately, a method for determining the speed of the system dynamics in real-time would be beneficial. Then the window size could be adjustable under different conditions and hopefully make the SDI more accurate and timely.
Chapter 5 Summary

5.1 Conclusions

The SDI has proven to be a valuable and necessary tool for assessing stability of the motion of NLTV systems. A number of difficulties had to be overcome in order to ensure the SDI provides accurate and timely results. A literature survey was conducted to show a need for a good stability assessment algorithm for NLTV systems and that the Lyapunov exponent is used in various fields for a wide array of problems. In addition, a number of tests were conducted to show, that with the addition of a few complimentary algorithms such as the Dead Band and a low pass filter, the SDI does provide accurate and timely results.

It is clear that, in this application, the use of a short-time Lyapunov exponent (SDI) is more applicable than the long-time Lyapunov exponent (LDI). This is due to the fact the RLV is a highly nonlinear dynamic system. To consider the long-time stability of this type of system is impractical and unnecessary. Therefore, the SDI is the preferred algorithm for this application.

It has been shown that a Lyapunov exponent that is mostly negative corresponds to a system that is converging to equilibrium and the negativity of the exponent is not necessary for the entire trajectory. It is acceptable for the vehicle's trajectory to diverge from the commanded trajectory provided it does not do so too quickly or by too large a magnitude. The SDI is able to distinguish an acceptable divergence from a critical one in a timely and accurate manner. This was proven by the results of the first two sets of tests.

Furthermore, the performance of the SDI is dependent on the choice of the window size $T$. For a system with fast dynamics, a shorter window size is required to ensure accurate and timely instability detection. Conversely, for a system with slow dynamics, a longer window size is more suitable. If the window sizes are chosen incorrectly, false alarms, missed detections and reductions in timeliness may be increased.
In the application of the RLV, it is beneficial to have different $T$ windows for not only the inner and outer loops, but also each channel. This is because each channel operates at different speeds and the ability to choose the proper window size would most likely enhance the SDI's performance. Ultimately, the window size should vary depending on the system's current dynamic rates throughout the vehicle's mission.

The well known method of gain and phase margins for assessing the robustness of linear systems has been successfully extended to the robustness assessment of nonlinear systems. Tests were conducted that clearly showed a close correlation between the gain and delay margins of a nonlinear system and its degree of robustness. Those cases where the gain and delay margins were small resulted in more failures than those cases with larger gain and delay margins. This showed that robustness of a nonlinear system can be enhanced by increasing the gain and delay margins. However, the increase in these margins negatively effects the nonlinear system's performance. In this application, the tracking errors were greater for cases with larger margins, while the cases with smaller margins showed very tight tracking. However, under perturbations, the system with larger margins was able to handle the perturbations and continue with the simulation while the converse was true for the smaller margin systems. Thus, there is likely a point at which the margins can be too large and result in poor performance and possibly an increase in the number of failed simulations.

A number of supporting algorithms were introduced to enhance the performance of the SDI. Among these algorithms are the Dead Band and the Butterworth low pass filter. The Dead Band solves the issue of the existence of a singular solution in the SDI, however, it introduces error into the solution. While this error can be considered negligible in some cases, it is undesired nonetheless. If the tracking error varies with a very small magnitude, this error could become unacceptable. Therefore, a more accurate method is desired and is available via the Hilbert Transformation. The low pass filter
proved to be beneficial as it reduced the amount of noise in the tracking error. The filter also introduces error into the system and this error could be unacceptable for tracking errors that have high frequencies. The filter is necessary and the filter parameters can be chosen to keep the error introduced into the solution minimal.

5.2 Future Work

While the SDI has proven to be a very effective means for assessing the stability of the motion of NLTV systems, there are still a number of items that need further exploration. A few supporting algorithms have been developed to enhance the capabilities of the SDI as an instability indicator. However, these algorithms introduce error into the calculation of the SDI and act as a suitable place holder for now. In addition, there are still further enhancements that can be applied or developed to further enhance this algorithm.

The Dead Band, which is used to avoid the possibility of the SDI resulting in a singular solution could be replaced by the Hilbert Transformation. If the tracking error is an analytic signal, the HT could be used to calculate the imaginary part from the real part (available tracking error) of the tracking error. When applied to the SDI, this will eliminate the possibility of a singular solution. There are a number of issues with this method, namely, the restriction that the tracking error must be analytic and the fact the HT is defined for infinite time. The use of a band pass filter can assure the tracking error is analytic, but the effects of implementing a finite time HT are unknown.

Since the SDI only calculates the Lyapunov exponent for the slowest mode, the ability to identify all modes could further increase the accuracy and promptness of the SDI. The Hilbert-Huang Transformation identifies all modes and applies the HT for each mode. The method for identifying these modes is not of use for this application, but it could be accomplished by using a bank of band pass filters tuned to different frequencies
to pinpoint these modes. The HT and the SDI could then be applied to each mode to obtain every Lyapunov exponent present in the system. This idea is being explored for other purposes in this research and will likely be feasible in the near future.

The short-time window for the SDI has been shown to have great effect on the algorithm's performance. This thesis presented an experimental method for determining the proper size window. However, this method is time consuming, tedious and does not allow for the calculation of a suitable window during the vehicle's mission. It is desirable to determine an analytic method for obtaining the proper $T$ value at varying points along the vehicle's trajectory. This could be done by analytically relating the time window to the speed of the vehicle's dynamics.

The method of gain and phase margins has been extended to NLTV systems and has proven to be a valuable method for determining the robustness of NLTV systems. An experimental method for determining these margins was carried out in this research. This method was tedious and time consuming as it made use of Monte Carlo dispersions. An analytic method for determining these margins would be beneficial.

Ultimately, the goal for developing the SDI is to eliminate the use of Monte Carlo dispersion tests as they are time consuming and tedious to analyze. For now, the SDI makes the results of these dispersion tests more meaningful and easier to analyze. The further development and study of the SDI might eventually allow developers and researchers to avoid this process all together.
References


Appendix A: SDI Program Code in MAVERIC/SLIM

/* Declarations for DI Program Code in afcs.c (Ascent controller)*/
/* Departure Index (Best 08/02/01) */
/* Modifications to pointer implementation  (Best 09/13/01) */

static double epsilon = .001; /* Dead Band bound */
double divisor; /* divisor in discrete integral computation */
static double current_time = 0; /* Time variable */
static vec short_DI_gamma; /* short DI variable of outer loop */
static vec short_DI_omega; /* short DI variable of inner loop */

double *long_DI_gamma_x; /* long term DI variable */
double *long_DI_gamma_y; /* long term DI variable */
double *long_DI_gamma_z; /* long term DI variable */
double *long_DI_omega_x; /* long term DI variable */
double *long_DI_omega_y; /* long term DI variable */
double *long_DI_omega_z; /* long term DI variable */
static int SHORT_STEP = 250; /* window size for short term DI */

/* End of Departure Index declarations */
/* End declarations for DI Program Code in afcs.c (Ascent controller)*/

/* Initializations for DI Program Code in afcs.c (Ascent controller)*/
/* Departure Index initializations */

long_DI_omega_x = (double *)malloc(SHORT_STEP * sizeof(double));
long_DI_omega_y = (double *)malloc(SHORT_STEP * sizeof(double));
long_DI_omega_z = (double *)malloc(SHORT_STEP * sizeof(double));
long_DI_gamma_x = (double *)malloc(SHORT_STEP * sizeof(double));
long_DI_gamma_y = (double *)malloc(SHORT_STEP * sizeof(double));
long_DI_gamma_z = (double *)malloc(SHORT_STEP * sizeof(double));

for (i = 0; i <= SHORT_STEP; ++i) {
    long_DI_gamma_x[i] = 0;
    long_DI_gamma_y[i] = 0;
    long_DI_gamma_z[i] = 0;
    long_DI_omega_x[i] = 0;
    long_DI_omega_y[i] = 0;
    long_DI_omega_z[i] = 0;
}

short_DI_gamma.x = 0;
short_DI_gamma.y = 0;
short_DI_gamma.z = 0;
short_DI_omega.x = 0;
short_DI_omega.y = 0;
short_DI_omega.z = 0;

/* End of Departure Index initializations */
/* End initializations for DI Program Code in afcs.c (Ascent controller)*/
/* Function calls for DI Program Code in afcs.c (Ascent controller)*/
/* Begin Departure Index code */

current_time += dtControl;  /* Total time */

/* Call to DI function */
short_DI_gamma = Departure_ID(smoothed_gamma, prev_smoothed_gamma, epsilon,
short_DI_gamma, current_time, dtControl, long_DI_gamma_x,
long_DI_gamma_y, long_DI_gamma_z, SHORT_STEP);

/* Call to DI function */
short_DI_omega = Departure_ID(smoothed_omega, prev_smoothed_omega, epsilon,
short_DI_omega, current_time, dtControl, long_DI_omega_x,
long_DI_omega_y, long_DI_omega_z, SHORT_STEP);

/* These statements moved to have access to prev_tracking error in DI code */
gamma_err_past = gamma_err;
omega_err_past = omega_err;
/* End Departure Index code */
/* End of function calls for DI Program Code in afcs.c (Ascent controller)*/

/* DI Program Code in stability.c */
/* Begin Departure Index code */

vec Departure_ID(vec smooth_error, vec prev_smooth_error, double epsilon, vec
windowed_DI, double total_time, double time_step, double
*DI_mem_x, double *DI_mem_y, double *DI_mem_z, int WINDOW)
{
    int i;  /* Increment variable for the FOR loop */
double prev_int[3];  /* Previous integral for current integral of x, y and z channels */
double DI_array[3];  /* Long DI for x, y and z channels */
double smooth_error_array[3];  /* Tracking error array for x, y and z channels */
double integral_array[3];  /* Integral array for x, y and z channels */
double prev_smooth_error_array[3]; /* Prev track error for x, y and z channels */
double windowed_DI_array[3]; /* Last value of DI array for x, y and z channels */
double delayed_int[3]; /* Delay integral for calculating SDI for x, y and z channels */
double delayed_DI_array[3]; /* Delayed value of DI, last value in DI_mem for x, y and z channels */
double windowed_int[3]; /* Integral for the short term window array for x, y and z channels */
double temp1, temp2; /* Used to get the absolute value of track errors */

/* Initialization of array variables */

DI_array[0] = DI_mem_x[0]; /* x channel DI, first value in the memory DI */
DI_array[1] = DI_mem_y[0]; /* y channel DI, first value in the memory DI */
DI_array[2] = DI_mem_z[0]; /* z channel DI, first value in the memory DI */
delayed_DI_array[0] = DI_mem_x[WINDOW]; /* x channel delayed DI, this is last value in the memory DI */
delayed_DI_array[1] = DI_mem_y[WINDOW]; /* y channel delayed DI, this is last value in the memory DI */
delayed_DI_array[2] = DI_mem_z[WINDOW]; /* z channel delayed DI, this is last value in the memory DI */
smooth_error_array[0] = smooth_error.x; /* New tracking error for x channel, needed to calculate integral */
smooth_error_array[1] = smooth_error.y; /* New tracking error for y channel, needed to calculate integral */
smooth_error_array[2] = smooth_error.z; /* New tracking error for z channel, needed to calculate integral */
prev_smooth_error_array[0] = prev_smooth_error.x; /* Previous tracking error for x channel, needed to calculate integral */
prev_smooth_error_array[1] = prev_smooth_error.y; /* Previous tracking error for y channel, needed to calculate integral */
prev_smooth_error_array[2] = prev_smooth_error.z; /* Previous tracking error for y channel, needed to calculate integral */

for (i = 0; i < 3; ++i) /* This loops for each channel; x, y, z to calculate the integral and long and short DI */
{
    /* Calculate previous integral from the previous DI and previous time */
    prev_int[i] = DI_array[i] * (total_time - time_step);
}
/* calculate delayed int from prev DI and prev time */
delayed_int[i] = delayed_DI_array[i] * (total_time - (time_step * WINDOW));
/* get absolute values for dead zone calc */
temp1 = absolute(smooth_error_array[i]);
/* get absolute values for dead zone calc */
temp2 = absolute(prev_smooth_error_array[i]);
if((temp1 < epsilon) && (temp2 < epsilon)) /* Dead band zone */
{
    /* If error = 0, just assign prev_int since zero would add nothing to integral */
    integral_array[i] = prev_int[i];
}
else
{
    /* absolute value of x_dot/x if have to divide by epsilon */
    if (smooth_error_array[i] > 0)
    {
        /* If error = 0, divide by epsilon, avoid division by zero */
        if (smooth_error_array[i] < epsilon)
        {
            integral_array[i] = prev_int[i] + (epsilon - prev_smooth_error_array[i]) / epsilon;
        }
        else /* error > epsilon and will not divide by zero */
        {
            integral_array[i] = prev_int[i] + (smooth_error_array[i] - prev_smooth_error_array[i]) / smooth_error_array[i];
        }
    }
    /* want absolute value of x_dot/x if have to divide by epsilon */
    else if (smooth_error_array[i] < 0)
    {
        /* want absolute value of x_dot/x so negate epsilon */
        if (smooth_error_array[i] > -epsilon)
        {
            integral_array[i] = prev_int[i] + (-epsilon - prev_smooth_error_array[i]) / -epsilon;
        }
        else /* error < epsilon and will not divide by zero */
        {
            integral_array[i] = prev_int[i] + (smooth_error_array[i] - prev_smooth_error_array[i]) / smooth_error_array[i];
        }
    }
}
\*\* integral delayed integral = windowed int \*/

\* windowed-int\[i\] = integral-array\[i\] - delayed-int\[i\];

\* short DI is the windowed int over the delay time in sec \*
\* windowed-DI-array\[i\] = windowed-int\[i\] \* (time-step \* WINDOW);

if (total-time < 0.2)
\* Reduce the transient in integral \*
{
    DI-array\[i\] = integral-array\[i\] / 0.2;
}
else
\* Long DI is the integral over time \*
DI-array\[i\] = integral-array\[i\] / total-time;

\* delay DI for short DI x channel \*
delay(D1\_mem\_x, DI-array\[0\], WINDOW);
windowed\_D1\_x = windowed\_DI-array\[0\];

\* short DI x channel \*
windowed\_D1\_x = windowed\_D1\_x / WINDOW;

\* short DI y channel \*
delay(D1\_mem\_y, DI-array\[1\], WINDOW);
windowed\_D1\_y = windowed\_DI-array\[1\];

\* short DI y channel \*
delay(D1\_mem\_z, DI-array\[2\], WINDOW);
windowed\_D1\_z = windowed\_DI-array\[2\];

\* short DI z channel \*

\* return windowed DI \*
return windowed\_D1;

\* DI Program Code in stability.c \*/

\* Return short DI \*/

\* integral delayed integral = windowed int \*
\* windowed\_int\[i\] = integral-array\[i\] - delayed-int\[i\];

\* short DI is the windowed int over the delay time in sec \*
\* reduce the transient in integral \*
if (total-time < 0.2)
{
    DI-array\[i\] = integral-array\[i\] / 0.2;
}
else
DI-array\[i\] = integral-array\[i\] / total-time;

\* delay DI for short DI x channel \*
delay(D1\_mem\_x, DI-array\[0\], WINDOW);
windowed\_D1\_x = windowed\_DI-array\[0\];

\* short DI x channel \*
windowed\_D1\_x = windowed\_D1\_x / WINDOW;

\* short DI y channel \*
delay(D1\_mem\_y, DI-array\[1\], WINDOW);
windowed\_D1\_y = windowed\_DI-array\[1\];

\* short DI y channel \*
delay(D1\_mem\_z, DI-array\[2\], WINDOW);
windowed\_D1\_z = windowed\_DI-array\[2\];

\* short DI z channel \*

\* return windowed DI \*
return windowed\_D1;
Appendix B: IDOS Implementation

(a) SDI for all 6 Channels

(b) SDI for Outer-Loop Roll Channel

(c) Low-pass Filter / Pseudo-Differentiator

(d) Short DI
Appendix C: Butterworth Filter Program Code in MAVERIC/SLIM

/***************************************************************************/
/* Declarations for Butterworth Program Code */

static vec s_gamma; /* Used to smooth the tracking errors */
static vec p_s_gamma;
static vec pp_s_gamma;
static vec s_omega; /* Used to smooth the tracking errors */
static vec p_s_omega;
static vec pp_s_omega;
vec smoothed_gamma;
vec smoothed_omega;
vec prev_smoothed_gamma;
vec prev_smoothed_omega;
vec smoothed; /* The smoothed tracking error */

/***************************************************************************/
/* End Declarations for Butterworth Program Code */

/*******************************************************************************/
/* Initializations for Butterworth Program Code */
/* Based on 1 Hz, or 6.28 rad/sec. Op bandwidth 2 rad/sec, Nyquist frq 25 Hz,
 filter bandwidth 4 rad/sec */

const double a0 = 1.0;
const double a1 = -1.9471;
const double a2 = 0.94857;

const double b0 = 0.097768;
const double b1 = -0.19403;
const double b2 = 0.097768;
/* End tracking error smoothing variables */

    s_gamma = Set_v(0., 0., 0.);
    p_s_gamma = Set_v(0., 0., 0.);
    pp_s_gamma = Set_v(0., 0., 0.);
    s_omega = Set_v(0., 0., 0.);
    p_s_omega = Set_v(0., 0., 0.);
    pp_s_omega = Set_v(0., 0., 0.);
smoothed_gamma = Set_v(0., 0., 0.);
smoothed_omega = Set_v(0., 0., 0.);

prev_smoothed_gamma = Set_v(0., 0., 0.);
prev_smoothed_omega = Set_v(0., 0., 0.);

/* End Initializations for Butterworth Program Code */

/* Calls to Butterworth and SDI Program Code */

s_gamma = smoothing(gamma_err, s_gamma, p_s_gamma, pp_s_gamma);
smoothed_gamma = smoothing_cont(s_gamma, p_s_gamma, pp_s_gamma);

pp_s_gamma = p_s_gamma;
p_s_gamma = s_gamma;

short_DI_gamma = Departure_ID(smoothed_gamma, prev_smoothed_gamma, epsilon,
short_DI_gamma, current_time, dtControl, long_DI_gamma_x, long_DI_gamma_y, long_DI_gamma_z, SHORT_STEP);
/* Call to DI function */

prev_smoothed_gama = smoothed_gama;

s_omega = smoothing(omega_err, s_omega, p_s_omega, pp_s_omega);
smoothed_omega = smoothing_cont(s_omega, p_s_omega, pp_s_omega);

pp_s_omega = p_s_omega;
p_s_omega = s_omega;

short_DI_omega = Departure_ID(smoothed_omega, prev_smoothed_omega, epsilon,
short_DI_omega, current_time, dtControl, long_DI_omega_x, long_DI_omega_y, long_DI_omega_z, SHORT_STEP);
/* Call to DI function */

prev_smoothed_omega = smoothed_omega;
/* End Calls to Butterworth and SDI Program Code */

/* Program Code */
vec smoothing(vec err, vec s, vec p_s, vec pp_s)
{
vec temp;
temp = Add_v(err, Mult_v_s(p_s, -a1));
return(Add_v(temp, Mult_v_s(pp_s, -a2)));
vec smoothing_cont(vec s, vec p_s, vec pp_s) /* Smooth tracking error for DI */
{
    vec temp;
    temp = Add_v(Mult_v_s(s, b0), Mult_v_s(p_s, b1));
    return(Add_v(temp, Mult_v_s(pp_s, b2)));
}
/* End Program Code */