49402414

GAIN SCHEDULING VIA CONTROL SIGNAL INTERPOLATION: THE BALL AND
BEAM EXAMPLE

A Thesis Presented to

The Faculty of the

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

Ohio University

In Partial Fulfillment

Of the Requirement for the Degree

Master of Science

by

Rouslan Ignatov

March, 2002

Thesis
m
2002
IGNA

OHIO UNIVERSITY
LIBRARY

2011/11/15 3/27/02

TABLE OF CONTENTS

List of Tables	v
List of Figures	vi
1. Introduction.....	1
1.1 Gain Scheduling Overview	1
1.2 History of Gain Scheduling	3
1.3 Current Directions.....	6
2. Gain scheduling theory	9
2.1 Gain Scheduling Fundamentals	9
2.2 Hidden Coupling.....	14
2.3 Velocity Implementation	20
2.4 Modified Velocity Algorithm (Modified D-Method).....	25
2.5 Control Signal Interpolation	28
2.6 Combining the D Method and Control Signal Interpolation	33
3. The Ball and Beam Example	37
3.1 System Description	37
3.2 Controller Design.....	40
3.3 Simulation Scheme	43
3.4 Performance Analysis	45
4. Concluding remarks.....	57
References.....	59
Appendix A: Matlab plant model (NLPlant.m)	61
Appendix B: Linear controller calculation (control.m)	63

Appendix C: System initialization (init.m).....	66
Appendix D: Simulation procedure (run.m)	68

LIST OF TABLES

Table 3.1: System variables and their physical equivalents	38
Table 3.2: Ball and Beam Parameters.....	38

LIST OF FIGURES

Figure 2.1 – Closed-loop step response	17
Figure 2.2 – Nonlinear simulation: step response ($r(t) = 0.2$)	18
Figure 2.3 – Nonlinear simulation: step response ($r(t) = 0.4$)	18
Figure 2.4 – Nonlinear simulation: step response ($r(t) = 0.6$)	19
Figure 2.5 – Structure of an interpolated controller	29
Figure 2.6 – The original controller architecture	33
Figure 2.7 – Modified controller architecture	35
Figure 3.1 - The ball and beam system	37
The equations of motion are given by	38
Figure 3.2 – Difference in tracking error response for a unit step input	42
Figure 3.3 – Difference in frequency response	43
Figure 3.4 – Simulation Diagram	44
Figure 3.5 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 1.5 m/s: linear controller	47
Figure 3.6 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 0.5 m/s.	47
Figure 3.7 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 0.5 m/s.	48
Figure 3.8 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 1.0 m/s.	48

Figure 3.9 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 1.0 m/s.	49
Figure 3.10 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 1.5 m/s.	49
Figure 3.11 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 1.5 m/s.	50
Figure 3.12 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 2.0 m/s.	50
Figure 3.13 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.0 m/s.	51
Figure 3.14 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 2.2 m/s.	51
Figure 3.15 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.2 m/s.	52
Figure 3.16 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp- and-hold slope of 2.5 m/s.	52
Figure 3.17 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.5 m/s.	53
Figure 3.18 - Transient response for controller switch from -0.5 m to -0.4 for ramp-and- hold with slope of 1.5 m/s	53
Figure 3.19 - Transient response for controller switch from 0.2 m to 0.3 for ramp-and- hold with slope of 1.5 m/s	54
Figure 3.20 - Transient response for controller switch from 0 m to 0.1 m for ramp-and- hold with slope of 2.2 m/s.	55
Figure 3.21 – Proposed controller vs. controller with analytical gain scheduling: Ball Position for ramp-and-hold with slope of 1.5 m/s.	56

Figure 3.22 – Proposed controller vs. controller with analytical gain scheduling: Beam

Angle for ramp-and-hold with slope of 1.5 m/s 56

Chapter 1

INTRODUCTION

1.1 Gain Scheduling Overview

Gain scheduling has recently gained increased attention as a formal nonlinear control system design method. It allows performance objectives to be met over a wide range of parameter variations while taking advantage of the wealth of tools and experience for linear controller synthesis. Gain scheduling has proven to be successful in different application domains such as flight control, chemical process control, and automotive control.

The design process first involves parameterizing plant equilibria and linearized models by designated scheduling variables and constructing an associated family of linear controllers. At this phase of the design process the scheduling variables are treated as frozen parameters that define the particular linearized plant and linear controller. The next step is the realization of a gain scheduled controller that continuously updates controller parameters based on measurements of the scheduling variables. Gain scheduled controllers usually have a linear parameter varying (LPV) structure. Here it is often implied that the gain scheduled controller automatically possesses an important linearization property: the linearization of the gain scheduled controller about any

equilibrium exactly matches the linear controller designed for this equilibrium. Often, the linearized controller does not exactly match the linear controller designed for a given equilibrium due to nonlinear dependence on the scheduling variables. Any such mismatch constitutes a so-called hidden coupling term that can potentially degrade and even destabilize system performance [1].

Exponential stability of the nonlinear closed-loop system about any equilibrium can be guaranteed by exponential stability of the linearized closed-loop system for this equilibrium, but only in a neighborhood of this equilibrium. This basic result cannot be applied to gain scheduled controllers directly because they have to provide system functionality over a wide range of operating conditions. The stability results that address this problem are given in [2], [3] and [4]. The main idea is that the closed-loop system has to be slowly varying and its initial state has to be sufficiently close to some equilibrium. Then the closed-loop system remains close to the equilibrium manifold.

The so-called velocity algorithm (D method) described in the literature [5], [6] provides a design technique yielding gain scheduled controllers that possess the linearization property stated above while retaining LPV structure. However, this method requires the availability of time-derivatives of certain signals, which is often impossible in practice. In this thesis, a gain scheduled controller implementation utilizing control signal interpolation that possesses the advantages of the D method but avoids the need for time-derivatives as controller inputs is proposed.

1.2 History of Gain Scheduling

There are not a lot of publications on gain scheduling before 1990. For example, in the 1963-1995 cumulative index, *Automatica* lists only four publication in this area, but only one of them is dated before 1990 [7]. There are several reasons for this situation. First of all gain scheduling was considered an ‘applied’ topic. Also, gain scheduled techniques and especially their hardware implementations often were proprietary or classified as military secrets. Because of cost considerations and performance requirements, the first implementation of gain scheduling techniques appeared in military applications. The jump start for this research was given by World War II and the development of new jet aircraft. In commercial applications, gain scheduling began when the digital computer control was introduced. We will take a closer look at the development of gain scheduled control in two application areas: flight control and automotive control.

Gain scheduling in autopilots is motivated by new generations of jet aircrafts and guided missiles with increased operating envelopes and more demanding performance requirements. The biggest engineering challenge arises from the fact that autopilot gains are highly dependent on variations in control surface effectiveness due to altitude and Mach number effects. So the main purpose of scheduling was to preserve constant loop-gain in spite of these influences. Most of the early implementations were designed to change gains only when flight conditions changed sufficiently. Attempts to vary the gains

in a continuous manner usually caused a failure rate to become unacceptable because of the complexity of the required hardware.

Early autopilot designs often reverted to a nominal constant gain in case of scheduling hardware failure. Also, redundancy was used widely. For example, separate airspeed measurements were used in individually gain scheduled, redundant controllers. Coupling between separate measurements and inaccuracies in the analog hardware became the prevalent problem in failure-tolerant systems, especially since the authority of the autopilot had increased. This provided the demand for rapid adoption of digital autopilot systems.

Here we describe some interesting features of gain-scheduled autopilots for a few well-known vehicles.

In the B-52 autopilot (Sperry Gyroscope Co.), developed in 1951, an airspeed-based mechanism was used to adjust the gain of the servo actuator's position feedback transducer. Conceptually this was model following control with an architecture that provided different transfer functions for the command and disturbance inputs. Loop gain was scheduled to maintain desired closed-loop bandwidth while surface effectiveness changed with flight condition. This approach avoided the need to schedule several variables.

The Nike Ajax missile (Bell Telephone Laboratories and Douglas Aircraft Co.), dated back to 1951, incorporated hardware gain scheduling on dynamic pressure based on a ram air tube that drove a multiple winding potentiometer set. One potentiometer was used to control a fin control system feedback gain and another one was responsible for

the gain of the roll control system. The major challenge in this system was hardware design because of a limited space in the control system compartment.

The Talos missile (Bendix Aviation, John Hopkins Applied Physics Laboratory) of the early 1950s used several scheduling schemes during different phases of operation. During the boost phase, controller gains were adjusted by a timer and then scheduled on altitude during the beam riding phase. Finally, on the terminal homing phase, gains were adjusted based on the difference between commanded and produced lateral acceleration to maintain good response. This type of control system was called a “sensitivity feedback system”, and it allowed simultaneous compensation of changes in altitude and speed.

In the automotive industry, the first examples of gain scheduled controllers for air/fuel mixture were developed in the early 1970s when the first microprocessors were developed. The precise control of air/fuel ratio became more important to meet new requirements on fuel economy and exhaust emissions.

In 1978, The Ford Motor Company presented its first production microprocessor based feedback control system. This system was used on Ford Pintos and Mercury Bobcats sold in California. The carburetor used a vacuum-actuated, variable orifice to meter fuel in response to the exhaust gas oxygen (EGO) sensor signal. Integral only control was used, and the gain (“ramp rate”) was dependent on engine speed. In the same year Ford also presented the first concept vehicle with fuel injection that used gain scheduling to control air/fuel ratio. General Motors presented its first system employing fuel injection and gain scheduling on engine speed and manifold pressure in 1982.

As digital microprocessors became more economical and powerful, this allowed more complex control systems to be implemented.

1.3 Current Directions

The major limitation of linearization gain scheduling is that stability properties of a closed-loop system can be guaranteed only in vicinity of the equilibrium manifold and under the assumption of slow variation of signals. Most of the research in the area of linearization gain scheduling is focused on developing techniques that allow these requirements to be relaxed.

In the case of rapidly changing signals, attempts have been made to use information about exogenous reference and scheduling signals in addition to their instantaneous values. Shamma and Cloutier [8] attempted to use the history of the scheduling variable while Sureshbabu and Rugh [9] implemented a scheme that required a derivative of the scheduling variable.

Another approach was developed by Tan and Rugh [10]. The objective of their work was to make linearization scheduling better address the nonlinear control context in which it is used. They tried to formulate linear controller point designs in such a way that they better contribute to the overall nonlinear objective. This work shows that linear-quadratic point designs must be related to each other so that at each equilibrium the gain scheduled controller approximates a so-called weak overtaking optimal control.

A similar idea is used as the basis for “stability-based switching” proposed by Leonessa Haddad and Chellaboina [11]. The main principle here is to select the operating conditions and to perform a linear point designs so that each operating condition is in the “capture region” of another operating condition. This restriction assures the stable transitions among operating conditions by appropriately switching controllers.

Gain scheduling usually involves the design of LPV controller for a LPV plant. Computational approaches for constructing quadratic parameter-dependent Lyapunov functions were the primary focus of LPV control design. As an alternative, Shamma and Xiong [12] and Tu and Shamma [13] considered the computational construction of nonquadratic Lyapunov functions. Another area of research is identification of LPV systems, illustrated by work of Bamieh and Giarré [14]. Also LPV tools can be applied for problems other than gain scheduling, for example controller saturation [15].

An important step in the gain scheduling process is the interpolation of the a set of linear point designs. The most recent research in this area is associated with a control signal interpolation technique [16], [17]. This interpolation method requires less computational power, especially for systems of the high order with multiple inputs outputs and scheduling variables. In contrast to interpolation of poles and zeros or state-space matrices, this technique applies no restrictions on the compensator architecture at each design point, because in this scheme the interpolation of the control signal occurs rather than controller parameters.

Gain scheduling offers the utility to construct a nonlinear control system using the well-developed linear design techniques, which include not only methods and algorithms

but also software and design tools. So the number of successful realizations of gain scheduled control systems will grow in the future. At the same time it is difficult to predict main directions in theoretical gain scheduling research. Most likely, new applications will give a strong driving force for theoretical inquiry and a test base for new ideas.

Chapter 2

GAIN SCHEDULING THEORY

2.1 Gain Scheduling Fundamentals

We consider a nonlinear continuous-time plant of the form

$$\begin{aligned}\dot{x}(t) &= f(x(t), w(t), u(t)) \\ z(t) &= h_z(x(t), w(t), r(t)) \\ y(t) &= h_y(x(t), w(t), r(t))\end{aligned}\tag{1}$$

where $x(t) \in R^n$ is the state vector, $w(t) \in R^{m_w}$ is a measured exogenous input used for scheduling, $u(t) \in R^{m_u}$ is the control input, $r(t) \in R^m$ is the reference input, $z(t) \in R^q$ is the regulated output, and $y(t) \in R^p$ is the measured output available for control and/or scheduling purposes. We assume that all functions are twice continuously differentiable.

Constant operating points, or equilibria, of the plant (1) for which the regulated variable is zero correspond to constant solutions of the system of algebraic equations

$$\begin{aligned}0 &= f(x^0, w^0, u^0) \\ 0 &= h_z(x^0, w^0, r^0) \\ y^0 &= h_y(x^0, w^0, r^0)\end{aligned}\tag{2}$$

We assume that this system of equations is underdetermined and has a family of solutions.

Assumption 2.1 For the nonlinear plant (1) we assume that for an l -dimensional vector of scheduling variables $\alpha = g(w, r, y)$, there exist an open set $0 \in \mathbf{O} \subset R^l$ and functions

$$\begin{aligned} x^0: \mathbf{O} &\rightarrow R^n, \quad w^0: \mathbf{O} \rightarrow R^{m_w}, \quad u^0: \mathbf{O} \rightarrow R^{m_u}, \\ r^0: \mathbf{O} &\rightarrow R^{m_r}, \quad y^0: \mathbf{O} \rightarrow R^p \end{aligned}$$

all zero at the origin, such that

$$\begin{aligned} 0 &= f(x^0(\alpha), w^0(\alpha), u^0(\alpha)) \\ 0 &= h_z(x^0(\alpha), w^0(\alpha), r^0(\alpha)) \\ y^0(\alpha) &= h_y(x^0(\alpha), w^0(\alpha), r^0(\alpha)) \end{aligned} \tag{3}$$

for $\alpha \in \mathbf{O}$.

The plant's parameterized family of equilibria induces a family of plant linearizations

$$\begin{aligned} \dot{x}_\delta(t) &= A(\alpha)x_\delta(t) + B_w(\alpha)w_\delta(t) + B_u(\alpha)u_\delta(t) \\ z_\delta(t) &= C_z(\alpha)x_\delta(t) + D_{zw}(\alpha)w_\delta(t) + D_{zr}(\alpha)r_\delta(t) \\ y_\delta(t) &= C_y(\alpha)x_\delta(t) + D_{yw}(\alpha)w_\delta(t) + D_{yr}(\alpha)r_\delta(t) \end{aligned} \tag{4}$$

where δ – subscripts indicate deviations from equilibria parameterized by the scheduling variable α , e.g., $x_\delta(t) = x(t) - x^0(\alpha)$, and the various coefficient matrices are given by the usual partial-derivative calculations on (1), for example,

$$A(\alpha) = \frac{\partial f}{\partial x}(x^0(\alpha), w^0(\alpha), u^0(\alpha)).$$

Suppose a parameterized family of linear controllers of the form

$$\begin{aligned} \dot{x}_{c\delta}(t) &= A_c(\alpha)x_{c\delta}(t) + B_c(\alpha)y_\delta(t) \\ u_\delta(t) &= C_c(\alpha)x_{c\delta}(t) + D_c(\alpha)y_\delta(t) \end{aligned} \tag{5}$$

has been constructed such that performance objectives are met by the feedback interconnection of (2) and (3) pointwise in α . It is important to note that the linear controller family is also described in terms of deviation variables but in this case the equilibrium state function $x_C^0(\cdot)$ associated with $x_{C\delta}(t)$ has yet to be specified.

Of interest is the existence of a continuous-time gain scheduled controller

$$\begin{aligned}\dot{x}_C(t) &= a(x_C(t), y(t), \alpha(t)) \\ u(t) &= c(x_C(t), y(t), \alpha(t))\end{aligned}\tag{6}$$

that possesses the linearization property mentioned in the Introduction, stated formally as the following requirement.

Requirement 2.2 For the gain scheduled controller (6) there must exist a continuously differentiable function $x_C^0(\cdot)$ satisfying

$$\begin{aligned}0 &= a(x_C^0(\alpha), y^0(\alpha), \alpha) \\ u_C^0(\alpha) &= c(x_C^0(\alpha), y^0(\alpha), \alpha).\end{aligned}\tag{7}$$

In addition, the following partial derivative identities must hold

$$\begin{aligned}\frac{\partial a}{\partial x_C}(x_C^0(\alpha), y^0(\alpha), \alpha) &= A_C(\alpha), & \frac{\partial c}{\partial x_C}(x_C^0(\alpha), y^0(\alpha), \alpha) &= C_C(\alpha), \\ \frac{\partial a}{\partial y}(x_C^0(\alpha), y^0(\alpha), \alpha) &= B_C(\alpha), & \frac{\partial c}{\partial y}(x_C^0(\alpha), y^0(\alpha), \alpha) &= D_C(\alpha), \\ \frac{\partial a}{\partial \alpha}(x_C^0(\alpha), y^0(\alpha), \alpha) &= 0, & \frac{\partial c}{\partial \alpha}(x_C^0(\alpha), y^0(\alpha), \alpha) &= 0.\end{aligned}\tag{8}$$

The first part of Requirement 2.2 ensures that the nonlinear closed-loop system has a family of equilibria for which the regulated variable is zero. The second part guarantees that the gain scheduled controller has a family of linearizations that agrees

with linear controller family (5) and is thus free of hidden coupling terms. Consequently, the nonlinear closed-loop system has a family of linearizations that exactly matches the feedback interconnection of (4) and (5). The following theorem taken from [1] gives a necessary and sufficient existence condition for gain scheduled controllers satisfying the linearization requirement.

Theorem 2.3 Given the family of continuous-time linear controllers (5), there exists a gain scheduled controller (6) satisfying Requirement 2.2 if and only if there exists a continuously differentiable function $x_c^0(\cdot)$ satisfying the partial differential equation

$$\begin{aligned} A_c(\alpha) \frac{\partial x_c^0}{\partial \alpha}(\alpha) + B_c(\alpha) \frac{\partial y^0}{\partial \alpha}(\alpha) &= 0 \\ C_c(\alpha) \frac{\partial x_c^0}{\partial \alpha}(\alpha) + D_c(\alpha) \frac{\partial y^0}{\partial \alpha}(\alpha) &= \frac{\partial u^0}{\partial \alpha}(\alpha). \end{aligned} \tag{9}$$

Proof. Suppose the controller (6) satisfies Requirement 2.1. Then, (7) holds for $\alpha \in \mathbf{O}$.

Differentiating (7) with respect to α and using the relationships (8) gives (9).

Now suppose $x_c^0(\alpha)$ satisfies (9) for the linear controller family specified by (5).

Define a gain scheduled controller of the form (6) by using the coefficient functions in (5) and the controller operating point functions according to

$$\begin{aligned} a(x_c, y, \alpha) &= A_c(\alpha)[x_c - x_c^0(\alpha)] + B_c(\alpha)[y - y^0(\alpha)] \\ c(x_c, y, \alpha) &= C_c(\alpha)[x_c(t) - x_c^0(\alpha)] + D_c(\alpha)[y(t) - y^0(\alpha)] + u^0(\alpha) \end{aligned} \tag{10}$$

For this controller, equilibria conditions (7) are automatically satisfied at each $\alpha \in \mathbf{O}$.

After a careful differentiation and evaluation at equilibria we achieve the corresponding controller linearization family, presented by

$$\begin{aligned}
\dot{x}_{C\delta} &= A_C(\alpha)x_{C\delta} + B_C(\alpha)y_\delta - \left[A_C(\alpha)\frac{\partial x_C^0(\alpha)}{\partial \alpha} + B_C(\alpha)\frac{\partial y^0(\alpha)}{\partial \alpha} \right] \times \\
&\left[\frac{\partial g}{\partial w}(y^0(\alpha), r^0(\alpha), w^0(\alpha))w_\delta + \frac{\partial g}{\partial r}(y^0(\alpha), r^0(\alpha), w^0(\alpha))r_\delta + \right. \\
&\quad \left. \frac{\partial g}{\partial y}(y^0(\alpha), r^0(\alpha), w^0(\alpha))y_\delta \right] \\
u_\delta &= C_C(\alpha)x_{C\delta}^c + D_C(\alpha)y_\delta + \left[\frac{\partial u^0(\alpha)}{\partial \alpha} - C_C(\alpha)\frac{\partial x_C^0(\alpha)}{\partial \alpha} - D_C(\alpha)\frac{\partial y^0(\alpha)}{\partial \alpha} \right] \times \\
&\left[\frac{\partial g}{\partial w}(y^0(\alpha), r^0(\alpha), w^0(\alpha))w_\delta + \frac{\partial g}{\partial r}(y^0(\alpha), r^0(\alpha), w^0(\alpha))r_\delta + \right. \\
&\quad \left. \frac{\partial g}{\partial y}(y^0(\alpha), r^0(\alpha), w^0(\alpha))y_\delta \right]
\end{aligned} \tag{11}$$

Substituting in (9), we see that (11) agrees with (5) because terms in a square brackets in (11) vanish. Thus, if (9) is satisfied, then the linearization of (11) matches the linearization with fixed scheduling variable α .

Gain scheduled controllers arising from linear parameter-varying implementations of linear controller family (5)

$$\begin{aligned}
\dot{x}_C(t) &= A_C(\alpha(t))x_C(t) + B_C(\alpha(t))y(t) \\
u(t) &= C_C(\alpha(t))x_C(t) + D_C(\alpha(t))y(t)
\end{aligned} \tag{12}$$

often introduce hidden coupling terms corresponding to the scheduling variables because partial derivative terms with respect to α that arise in the linearization process are typically nonzero as required by the last two identities in (8).

2.2 Hidden Coupling

To analyze the origin of hidden coupling consider the following controller of the form

$$\begin{aligned}\dot{x}_c(t) &= A_c(\alpha(t))[x_c(t) - x_c^0(\alpha(t))] + B_c(\alpha(t))[y(t) - y^0(\alpha(t))] \\ u(t) &= C_c(\alpha(t))[x_c(t) - x_c^0(\alpha(t))] + D_c(\alpha(t))[y(t) - y^0(\alpha(t))] + u^0(\alpha(t))\end{aligned}\tag{13}$$

where

$$\alpha(t) = g(w(t), r(t), y(t))\tag{14}$$

For this controller, condition (7) is obviously satisfied.

To build a gain scheduled controller we replace the parameter α by its dependence on other variables according to (14).

This substitution leads to a gain scheduled controller resulting from the definition in (10) (omitting t-arguments)

$$\begin{aligned}\dot{x}_c(t) &= A_c(\alpha)[x_c - x_c^0(g(y, w))] + B_c(\alpha)[y - y^0(g(y, w))] \\ u(t) &= C_c(\alpha)[x_c - x_c^0(g(y, w))] + D_c(\alpha)[y - y^0(g(y, w))] + u^0(g(y, w))\end{aligned}\tag{15}$$

Linearization of (15) about an equilibrium specified by α gives equations (11). If the condition (9) does not hold then the resulting equations does not match (5). The additional terms that arise from this mismatch are usually called hidden coupling terms.

To demonstrate the influence that hidden coupling has on the behavior of a gain scheduled controller consider the following example [7].

Example 2.1. Consider the scalar nonlinear plant

$$\begin{aligned}\dot{x} &= -x + u, \\ y &= \tanh(x), \\ z &= r - y = r - \tanh(x).\end{aligned}\tag{16}$$

We assume the objective is that $y(t)$ track an exogenous reference signal $r(t)$ with acceptable transient response and zero steady-state error for constant $r(t)$. It is obvious that there exists an equilibrium family that corresponds to zero tracking error which can be parameterized by $\alpha \in (-1, 1)$ as

$$\begin{aligned}x^0(\alpha) &= u^0(\alpha) = \tanh^{-1}(\alpha), \\ r^0(\alpha) &= y^0(\alpha) = \alpha, \\ z^0(\alpha) &= 0.\end{aligned}\tag{17}$$

Deviation variables are defined as follows

$$\begin{aligned}x_\delta(t) &= x(t) - \tanh^{-1}(\alpha), \quad u_\delta(t) = u(t) - \tanh^{-1}(\alpha), \\ y_\delta(t) &= y(t) - \alpha, \quad r_\delta(t) = r(t) - \alpha, \quad z_\delta(t) = z(t).\end{aligned}$$

The plant dynamics can then be linearized about this constant operating point family to yield a family of linearizations

$$\begin{aligned}\dot{x}_\delta &= -x_\delta + u_\delta, \\ y_\delta &= (1 - \alpha^2)x_\delta, \\ z_\delta &= r_\delta - y_\delta.\end{aligned}\tag{18}$$

For the second step, we represent the family of linearized plants by the parameterized transfer function

$$G_\alpha^p(s) = \frac{Y_\delta(s)}{U_\delta(s)} = \frac{1 - \alpha^2}{s + 1}, \quad \alpha \in (-1, 1)\tag{19}$$

and consider a parameterized family of proportional-integral controllers defined by

$$G_{\alpha}^C(s) = \frac{Y_{\delta}(s)}{U_{\delta}(s)} = k_p(\alpha) + \frac{k_I(\alpha)}{s}. \quad (20)$$

This results in a family of closed-loop unity-feedback linearizations represented by

$$T_{\alpha}(s) = \frac{(1 - \alpha^2)[k_p(\alpha)s + k_I(\alpha)]}{s^2 + [1 + (1 - \alpha^2)k_p(\alpha)]s + (1 - \alpha^2)k_I(\alpha)} \quad (21)$$

It is clear that the PI gains can be chosen as functions of the scheduling variable in such a way that the family of closed-loop transfer functions is independent of the particular constant operating point parameterized by α .

For example, choosing

$$k_p(\alpha) = \frac{1}{3(1 - \alpha^2)}, \quad k_I(\alpha) = \frac{1}{(1 - \alpha^2)}$$

yields the closed-loop transfer function

$$T_{\alpha}(s) = \frac{Y_{\delta}(s)}{U_{\delta}(s)} = \frac{\frac{1}{3}s + 1}{s^2 + \frac{4}{3}s + 1} \quad (22)$$

The unit step response of this closed-loop system is presented in Figure 2.1

Next, select the following state-space realization of $G_{\alpha}^C(s)$

$$\begin{aligned} \dot{x}_{c\delta}(t) &= r_{\delta}(t) - y_{\delta}(t) \\ u_{\delta}(t) &= k_I(\alpha)x_{c\delta}(t) + k_p(\alpha)[r_{\delta}(t) - y_{\delta}(t)] \end{aligned} \quad (23)$$

Now, a traditional approach to implementing a gain scheduled controller from a family of linear designs would result in

$$\begin{aligned}\dot{x}_c(t) &= r(t) - y(t) \\ u_\delta(t) &= k_I(y(t))x_c(t) + k_P(y(t))[r(t) - y(t)]\end{aligned}\tag{24}$$

where dependence on frozen values of the scheduling variable, α , has been replaced by dependence on the instantaneous signal $y(t)$. Note that this implementation has constant

operating points specified by $r^0(\alpha) = \alpha$ and $x_c^0(\alpha) = \frac{u^0(\alpha)}{k_I(\alpha)} = [1 - \alpha^2] \tanh^{-1}(\alpha)$.

Figure 2.1 – Closed-loop step response

Nonlinear simulations were performed on the closed-loop system constructed by interconnection of the nonlinear plant and gain scheduled controller (24) for reference

step inputs of amplitude 0.2, 0.3 and 0.6. The resulting transient responses are presented on Figures 2.2-2.4.

Figure 2.2 – Nonlinear simulation: step response ($r(t) = 0.2$)

Figure 2.3 – Nonlinear simulation: step response ($r(t) = 0.4$)

Figure 2.4 – Nonlinear simulation: step response ($r(t) = 0.6$)

Nonlinear simulations show that the closed-loop system described by

$$\begin{aligned}
 \dot{x}(t) &= -x(t) + k_I(y(t))x_C(t) + k_P(y(t))[r(t) - y(t)] \\
 \dot{x}_C(t) &= r(t) - y(t) \\
 y(t) &= \tanh(x(t))
 \end{aligned} \tag{25}$$

becomes unstable when the reference step input $r(t) > 0.63$. This behavior is caused by the fact that the gain scheduled controller (24) does not linearize to the family of linear PI controllers (23) due to dependence of controller gains on $y(t)$, hence on $x(t)$. This dependence causes an additional destabilizing feedback term to appear in the family of closed-loop linearizations. Linearization of (25) about its family of constant operating points given by functions $x^0(\alpha), x_C^0(\alpha), r^0(\alpha)$ results in

$$\begin{aligned}
\begin{bmatrix} \dot{x}_\delta(t) \\ \dot{x}_{c\delta}(t) \end{bmatrix} &= \begin{bmatrix} -\frac{4}{3} + 2\alpha \tanh^{-1}(\alpha) & k_I(\alpha) \\ (1-\alpha^2) & 0 \end{bmatrix} \begin{bmatrix} x_\delta(t) \\ x_{c\delta}(t) \end{bmatrix} + \begin{bmatrix} k_P(\alpha) \\ 1 \end{bmatrix} r_\delta \\
y_\delta(t) &= \begin{bmatrix} (1-\alpha^2) & 0 \end{bmatrix} \begin{bmatrix} x_\delta(t) \\ x_{c\delta}(t) \end{bmatrix}
\end{aligned} \tag{26}$$

with associated family of closed-loop transfer functions

$$T_\alpha(s) = \frac{\frac{1}{3}s + 1}{s^2 + [\frac{4}{3} - 2\alpha \tanh^{-1}(\alpha)]s + 1} \tag{27}$$

Closed-loop linearizations are unstable when $2\alpha \tanh^{-1}(\alpha) > \frac{4}{3}$, occurring when $\alpha > 0.7254$.

2.3 Velocity Implementation

The requirement on a gain scheduled controller to satisfy the linearization property stated above can be relaxed. For a particular controller architecture, the linearization property could be stated as a set of two requirements. These are that at each equilibrium the linearized closed-loop, gain scheduled system should have the same eigenvalues and same transfer function as the linear closed-loop system composed of the linearized plant and designed linear controller [5]. For a typical tracking problem, this allows constructing a gain scheduled controller which is free of hidden coupling terms and simpler than the controller described by (15).

To illustrate, consider tracking a constant reference input $r(t)$ using an integral-error controller. The objective is to maintain a zero steady-state error, that is $y^0(\alpha) = r^0(\alpha)$. Assume that the regulated output is the part of the measured output, that is,

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad y_1 = z \quad (28)$$

This partitioning has the following impact on the plant constant operating point function

$$y^0(\alpha) = \begin{bmatrix} 0 \\ y_2^0(\alpha) \end{bmatrix}. \quad (29)$$

and the plant linearization family is now given by

$$\begin{aligned} \dot{x}_\delta(t) &= A(\alpha)x_\delta(t) + B_w(\alpha)w_\delta(t) + B_u(\alpha)u_\delta(t) \\ z_\delta(t) &= C_z(\alpha)x_\delta(t) + D_{zw}(\alpha)w_\delta(t) + D_{zr}(\alpha)r_\delta(t) \\ y_{2\delta}(t) &= C_{y2}(\alpha)x_\delta(t) + D_{yw2}(\alpha)w_\delta(t) + D_{yr2}(\alpha)r_\delta(t) \end{aligned} \quad (30)$$

We consider two linear controller families containing integral action. In first the regulated output is integrated directly leading to

$$\begin{aligned} \dot{x}_{C1\delta}(t) &= z_\delta(t) \\ x_{C2\delta}(t) &= A_{21}(\alpha)x_{C1\delta}(t) + A_{22}(\alpha)x_{C2\delta}(t) + B_{21}(\alpha)z_\delta(t) + B_{22}(\alpha)y_{2\delta}(t) \\ u_\delta(t) &= C_1(\alpha)x_{C1\delta}(t) + C_2(\alpha)x_{C2\delta}(t) + D_1(\alpha)z_\delta(t) \end{aligned} \quad (31)$$

where we have disallowed direct feedthrough from $y_{2\delta}$ to u_δ . In addition, we make the

following assumptions:

A.1 the feedback interconnection of the plant linearization family (30) and linear controller family (31) is asymptotically stable for each $\alpha \in \mathbf{O}$;

A.2 $m_u = q$;

The closed-loop system is specified by (dropping α - arguments)

$$\begin{bmatrix} \dot{x}_\delta \\ \dot{x}_{C1\delta} \\ \dot{x}_{C2\delta} \end{bmatrix} = \begin{bmatrix} A + B_u D_1 C_z & B_u C_1 & B_u C_2 \\ C_z & 0 & 0 \\ B_{21} C_z + B_{22} C_{y2} & A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_\delta \\ x_{C1\delta} \\ x_{C2\delta} \end{bmatrix} + \begin{bmatrix} B_w + B_u D_1 D_{zw} \\ D_{zw} \\ B_{21} D_{zw} + B_{22} D_{yw2} \end{bmatrix} w_\delta + \begin{bmatrix} B_u D_1 D_{zr} \\ D_{zr} \\ B_{21} D_{zr} + B_{22} D_{yr2} \end{bmatrix} r_\delta \quad (32)$$

From Assumption A.1, The closed-loop matrix

$$A_{CL}^1 = \begin{bmatrix} A + B_u D_1 C_z & B_u C_1 & B_u C_2 \\ C_z & 0 & 0 \\ B_{21} C_z + B_{22} C_{y2} & A_{21} & A_{22} \end{bmatrix} \quad (33)$$

is Hurwitz pointwise in α .

It is not difficult to show that this stability property holds only if

$$\begin{bmatrix} C_1(\alpha) & C_2(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} \quad (34)$$

has full column rank at each $\alpha \in \mathbf{O}$ and is therefore square and nonsingular by Assumption A.2.

To derive the second linear controller family, we begin with the parameterized transfer function from $(z_\delta, y_{\delta 2})$ to u_δ associated with (31) (ommiting α -arguments)

$$\begin{bmatrix} C_2(sI - A_{22})^{-1} A_{21} + C_1 & C_2(sI - A_{22})^{-1} B_{21} + D_1 & C_2(sI - A_{22})^{-1} B_{22} \end{bmatrix} \begin{bmatrix} \frac{1}{s} I & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \quad (35)$$

From an input/output stand point, this transfer function is equivalent to

$$\frac{1}{s} \begin{bmatrix} C_2(sI - A_{22})^{-1} A_{21} + C_1 & C_2(sI - A_{22})^{-1} B_{21} + D_1 & C_2(sI - A_{22})^{-1} B_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ sI & 0 \\ 0 & sI \end{bmatrix} \quad (36)$$

This manipulation corresponds to moving the integrators to the controller output and inserting differentiators on those controller inputs that were not originally integrated.

A state-space realization of this controller family is given by

$$\begin{aligned} \dot{\xi}_{C1\delta} &= C_2(\alpha)\xi_{C2\delta} + C_1(\alpha)z_\delta, \\ \dot{\xi}_{C2\delta} &= A_{22}(\alpha)\xi_{C2\delta} + A_{21}(\alpha)z_\delta + B_{21}(\alpha)\dot{z}_\delta + B_{22}(\alpha)\dot{y}_{2\delta}, \\ u_\delta &= \xi_{C1\delta} + D_1(\alpha)z_\delta \end{aligned} \quad (37)$$

with a corresponding closed-loop state equation family of the form (omitting α -arguments)

$$\begin{aligned} \begin{bmatrix} \dot{x}_\delta \\ \dot{\xi}_{C1\delta} \\ \dot{\xi}_{C2\delta} \end{bmatrix} &= \begin{bmatrix} A + B_u D_1 C_z & B_u & 0 \\ C_1 C_z & 0 & C_2 \\ A_{21} C_z + (B_{21} C_z + B_{22} C_{y2})(A + B_u D_1 C_z) & B_{21} C_z B_u + B_{22} C_{y2} & A_{22} \end{bmatrix} \begin{bmatrix} x_\delta \\ \xi_{C1\delta} \\ \xi_{C2\delta} \end{bmatrix} + \\ &\begin{bmatrix} B_w + B_u D_1 D_{zw} \\ C_1 D_{zw} \\ A_{21} D_{zw} + (B_{21} C_z + B_{22} C_{y2})(B_w + B_u D_1 D_{zw}) \end{bmatrix} w_\delta + \\ &\begin{bmatrix} B_u D_1 D_{zr} \\ C_1 D_{zr} \\ A_{21} D_{zr} + (B_{21} C_z + B_{22} C_{y2}) B_u D_1 D_{zw} \end{bmatrix} r_\delta + \\ &\begin{bmatrix} 0 \\ 0 \\ B_{21} D_{zw} + B_{22} D_{y2} \end{bmatrix} \dot{w}_\delta + \begin{bmatrix} 0 \\ 0 \\ B_{21} D_{zr} + B_{22} D_{yr2} \end{bmatrix} \dot{r}_\delta. \end{aligned} \quad (38)$$

Both controllers have the same transfer function from $(z_\delta, y_{2\delta})$ to u so the two closed-loop linear families have the same transfer function from (w_δ, r_δ) to $(z_\delta, y_{2\delta})$.

To show that both closed-loop linear systems have the same eigenvalues we need to find a similarity transformation that relates these two systems. So we need to find a square invertible matrix T such that

$$A_{CL}^1 = T^{-1} A_{CL}^2 T \quad (39)$$

$$\text{where } A_{CL}^1 = \begin{bmatrix} A + B_u D_1 C_z & B_u C_1 & B_u C_2 \\ C_z & 0 & 0 \\ B_{21} C_z + B_{22} C_{y2} & A_{21} & A_{22} \end{bmatrix}$$

$$\text{and } A_{CL}^2 = \begin{bmatrix} A + B_u D_1 C_z & B_u & 0 \\ C_1 C_z & 0 & C_2 \\ A_{21} C_z + (B_{21} C_z + B_{22} C_{y2})(A + B_u D_1 C_z) & B_{21} C_z B_u + B_{22} C_{y2} & A_{22} \end{bmatrix}$$

This requirement can be simplified using basic matrix properties to yield

$$A_{CL}^2 T = T A_{CL}^1 \quad (40)$$

It is not very difficult to show that this condition is satisfied by

$$T = \begin{bmatrix} I & 0 & 0 \\ 0 & C_1 & C_2 \\ B_{21} C_z + B_{22} C_{y2} & A_{21} & A_{22} \end{bmatrix} \quad (41)$$

which is nonsingular due to nonsingularity of (34). So the two closed-loop linear systems have the same eigenvalues.

The two linear systems have the same transfer functions and the same eigenvalues, thus the linearization property is satisfied. A gain scheduled controller corresponding to the linear controller family (37) is given by

$$\begin{aligned} \dot{x}_{c1}(t) &= C_2(\alpha(t))x_{c2}(t) + C_1(\alpha(t))z(t), \\ \dot{x}_{c2}(t) &= A_{22}(\alpha(t))x_{c2}(t) + A_{21}(\alpha(t))z(t) + B_{21}(\alpha(t))\dot{z}(t) + B_{22}\dot{y}_2(t) \\ u(t) &= x_{c1}(t) + D_1(\alpha(t))z(t). \end{aligned} \quad (42)$$

which has an LPV structure but requires the derivatives $\dot{z}$ and $\dot{y}$ as inputs.

2.4 Modified Velocity Algorithm (Modified D-Method)

In this section, we describe a gain scheduled controller implementation for which the existence condition of Theorem 2.3 is trivially satisfied. The technique is motivated by the D method illustrated in Section 2.3 in that integrators within the controller are judiciously relocated in such a way as to preserve the input/output behavior of the linear controller family, yet time-derivatives of controller inputs will not be required here. The basic idea is to absorb the differentiators that appear in the controller architecture of the D method into the controller dynamics [16]. This manipulation can be cast as a coordinate transformation on the linear controller family that preserves input /output behavior and allows (9) to be solved by inspection.

We start with the controller designed in the previous section. The controller transfer function is given by (36). We need to introduce another assumption on our linear controller family:

$$\mathbf{A.3} \quad B_{22}(\alpha) = B_{22}, \text{ constant.}$$

Next, using the identity

$$s(sI - A)^{-1} = I + (sI - A)^{-1} A \quad (43)$$

it is possible to absorb the differentiators back into the controller dynamics to yield

$$\begin{aligned} \frac{1}{s} \left[C_2(sI - A_{22})^{-1} (A_{21} + A_{22}B_1) + (C_1 + C_2B_{21}) \quad C_2(sI - A_{22})^{-1} A_{22}B_{22} + C_2B_{22} \right] \\ + [D_1 \quad 0]. \end{aligned} \quad (44)$$

A linear controller family of the form (5) corresponding to the transfer function (44) is given by

$$\begin{aligned}\dot{\psi}_{C1} &= C_2(\alpha)\psi_{C2} + (C_1(\alpha) + C_2(\alpha)B_{21}(\alpha))z_\delta + C_2(\alpha)B_{22}(\alpha)y_\delta \\ \dot{\psi}_{C2} &= A_{22}(\alpha)\psi_{C2} + (A_{21}(\alpha) + A_{22}(\alpha)B_{21}(\alpha))z_\delta + A_{22}(\alpha)B_{22}(\alpha)y_\delta \\ u_\delta &= \psi_{C2} + D_1(\alpha)z_\delta\end{aligned}\tag{45}$$

It is not difficult to show that two linear controller families are related by a parameterized linear coordinate transformation

$$\begin{aligned}\begin{bmatrix} 0 & C_2(\alpha) \\ 0 & A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} C_1(\alpha) & C_2(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} &= \begin{bmatrix} C_1(\alpha) & C_2(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} 0 & 0 \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} \\ \begin{bmatrix} C_1(\alpha) + C_2(\alpha)B_{21}(\alpha) & C_2(\alpha)B_{22}(\alpha) \\ A_{22}(\alpha) + A_{22}(\alpha)B_{21}(\alpha) & A_{22}(\alpha)B_{22}(\alpha) \end{bmatrix} &= \begin{bmatrix} C_1(\alpha) & C_2(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} I & 0 \\ B_{21}(\alpha) & B_{22}(\alpha) \end{bmatrix} \\ [I & 0] \begin{bmatrix} C_1(\alpha) & C_2(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} &= [C_1(\alpha) & C_2(\alpha)]\end{aligned}\tag{46}$$

Thus, the feedback interconnection of the plant linearization family (30) and the linear controller family (45) is asymptotically stable pointwise in α and has the same parameterized transfer function as the feedback interconnection of (30) and (31).

Next, we consider the implementation of a gain scheduled controller that satisfies Requirement 2.2 with respect to linear controller family (45). After partitioning

$$\begin{aligned}\psi_c^0(\alpha) &= \begin{bmatrix} \psi_{C1}^0 \\ \psi_{C2}^0 \end{bmatrix}, \\ y^0(\alpha) &= \begin{bmatrix} 0 \\ y_2^0 \end{bmatrix}.\end{aligned}\tag{47}$$

the identities of Theorem 2.3 can be rewritten as follows

$$\begin{aligned} \begin{bmatrix} 0 & C_2(\alpha) \\ 0 & A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} \frac{\partial \psi_{c1}^0}{\partial \alpha}(\alpha) \\ \frac{\partial \psi_{c2}^0}{\partial \alpha}(\alpha) \end{bmatrix} + \begin{bmatrix} C_1(\alpha) + C_2(\alpha)B_{21}(\alpha) & C_2(\alpha)B_{22}(\alpha) \\ A_{21}(\alpha) + A_{22}(\alpha)B_{21}(\alpha) & A_{22}(\alpha)B_{22}(\alpha) \end{bmatrix} \begin{bmatrix} 0 \\ \frac{\partial y_2^0}{\partial \alpha}(\alpha) \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ [I \quad 0] \begin{bmatrix} \frac{\partial \psi_{c1}^0}{\partial \alpha}(\alpha) \\ \frac{\partial \psi_{c2}^0}{\partial \alpha}(\alpha) \end{bmatrix} + [D_1(\alpha) \quad 0] \begin{bmatrix} 0 \\ \frac{\partial y_2^0}{\partial \alpha}(\alpha) \end{bmatrix} &= \frac{\partial u^0}{\partial \alpha}(\alpha) \end{aligned} \quad (48)$$

It is easy to see that these identities reduce to

$$\begin{aligned} \begin{bmatrix} C_2(\alpha) \\ A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} \frac{\partial \psi_{c2}^0}{\partial \alpha}(\alpha) + B_{22} \frac{\partial y_2^0}{\partial \alpha}(\alpha) \end{bmatrix} &= 0 \\ \frac{\partial \psi_{c2}^0}{\partial \alpha}(\alpha) &= \frac{\partial u^0}{\partial \alpha}(\alpha). \end{aligned} \quad (49)$$

Due to nonsingularity of (34), the leftmost factor in the first identity has full column rank, so the first identity can be further to

$$\frac{\partial \psi_{c2}^0}{\partial \alpha}(\alpha) + B_{22} \frac{\partial y_2^0}{\partial \alpha}(\alpha) = 0. \quad (50)$$

As a consequence of Assumption A.3 the first identity is satisfied by taking $\psi_{c2}^0(\alpha) = -B_{22}y_2^0(\alpha)$ and the second identity is satisfied by taking $\psi_{c1}^0(\alpha) = u^0(\alpha)$. Invoking Theorem 2.3, we see that it is possible to implement a gain scheduled controller of a general form (6) that satisfies Requirement 2.2 with respect to the linear controller family (45).

Such a controller is not uniquely specified, but a suitable choice is the following

$$\begin{aligned}\dot{\psi}_{C_1}(t) &= C_2(\alpha(t))\psi_{C_2}(t) + [C_1(\alpha(t)) + C_2(\alpha(t))B_{21}(\alpha(t))]z(t) + C_2(\alpha(t))B_{22}y_2(t) \\ \dot{\psi}_{C_2}(t) &= A_{22}(\alpha(t))\psi_{C_2}(t) + [A_{21}(\alpha(t)) + A_{22}(\alpha(t))B_{21}(\alpha(t))]z(t) + A_{22}(\alpha(t))B_{22}y_2(t) \\ u(t) &= \psi_{C_1}(t) + D_1(\alpha(t))z(t)\end{aligned}\quad (51)$$

This gain scheduled controller obviously has an LPV structure derived directly from the linear controller family (45).

2.5 Control Signal Interpolation

Linear controller families are often constructed from a discrete set of linear point designs via some interpolation scheme. Often, controller poles and zeros, transfer function coefficients, or state-space matrices are interpolated. Here we consider interpolation of the control signals generated by the linear point designs, which is equivalent to interpolating their transfer functions. For simplicity, we describe this process in the case of a scalar scheduling variable.

Let $\alpha^i, i=1, \dots, N$ denote a discrete set of scheduling variable values for which linear point designs represented by minimal realizations $(A_C^i, B_C^i, C_C^i, D_C^i)$ have been constructed. For $\alpha \in [\alpha^i, \alpha^{i+1})$, we linearly interpolate the control signals generated by the i^{th} and $(i+1)^{st}$ point designs which in the frequency domain corresponds to

$$U_\delta(s) = [p^i(\alpha)K^i(s) + p^{i+1}(\alpha)K^{i+1}(s)]Y_\delta(s) \quad (52)$$

where $K^i(s) = C_C^i(sI - A_C^i)^{-1}B_C^i + D_C^i$ and the interpolating functions are defined by

$$\rho^i(\alpha) = \frac{\alpha^{i+1} - \alpha}{\alpha^{i+1} - \alpha^i}, \quad \rho^{i+1}(\alpha) = \frac{\alpha - \alpha^i}{\alpha^{i+1} - \alpha^i}. \quad (53)$$

In state-space terms, the linear controller family (3) for $\alpha \in [\alpha^i, \alpha^{i+1})$ is given by

$$\begin{aligned} A_c(\alpha) &= \begin{bmatrix} A_c^i & 0 \\ 0 & A_c^{i+1} \end{bmatrix}, \quad B_c(\alpha) = \begin{bmatrix} B_c^i \\ B_c^{i+1} \end{bmatrix}, \\ C_c(\alpha) &= [\rho^i(\alpha)C_c^i \quad \rho^{i+1}(\alpha)C_c^{i+1}], \\ D_c(\alpha) &= \rho^i(\alpha)D_c^i + \rho^{i+1}(\alpha)D_c^{i+1}. \end{aligned} \quad (54)$$

The structure of the interpolated controller is presented in Figure 2.5. Note that the dimension of this parallel realization equals the sum of the dimensions of the two linear point designs. However, the dimension of the linear point designs need not be the same, as is required by other interpolation schemes.

Figure 2.5 – Structure of an interpolated controller

Another important property which this interpolation scheme possesses is that the controller's B matrix is always constant. Consequently the portion that represents B_{22} in

Assumption A.3 is constant. So, all the assumptions are satisfied and therefore the modified D method together with the proposed control signal interpolation scheme leads to construction of a gain scheduled controller that is free of hidden coupling terms.

A significant challenge that arises when implementing control signal interpolation has to do with the controller switching that occurs as $\alpha(t)$ passes from one interval to the next. In particular, a new controller that comes on-line must be initialized properly so as not to introduce an undesirable transient effect. Here we describe our approach to controller initialization. Let times t_{i-1} and t_i be such that $\alpha(t) \in [\alpha^{i-1}, \alpha^i)$ for $t \in [t_{i-1}, t_i)$ and at time t_i $\alpha(t)$ passes into the adjacent interval $[\alpha^i, \alpha^{i+1})$. In this scenario, the controller $K^i(s)$ remains in service during this transition and the controller $K^{i+1}(s)$ comes on-line. Our goal is to calculate an initial state for this controller that simulates a ‘warm-up’ period during the preceding time interval $[t_{i-1}, t_i)$. We assume that the only data available for this calculation are the states $x_C^i(t_{i-1})$ and $x_C^i(t_i)$. In particular, the input signal to the i^{th} controller, $y(t)$, $t \in [t_{i-1}, t_i)$, is not available, but can be approximated via

$$\hat{y}(t) = B_C^{iT} \exp \left[A_C^i (t - t_{i-1}) \right] W_i^{-1}(t_i, t_{i-1}) \cdot \left(x_C^i(t_i) - \exp \left[A_C^i (t_i - t_{i-1}) \right] x_C^i(t_{i-1}) \right) \quad (55)$$

where

$$W_i(t_i, t_{i-1}) = \int_{t_{i-1}}^{t_i} \exp[A_C^i(t_i - \tau)] \cdot B_C^i B_C^{iT} \cdot \exp[A_C^{iT}(t_i - \tau)] d\tau \quad (56)$$

denotes the Reachability Grammian for the pair (A_C^i, B_C^i) . Making the variable change

$\sigma = t_i - \tau$; $d\sigma = -d\tau$ we have the following equation for the Reachability Grammian

$$\begin{aligned}
W_i(t, t_{i-1}) &= \int_{t-t_{i-1}}^0 \exp(A_C^i \sigma) \cdot B_C^i B_C^{iT} \cdot \exp(A_C^{iT} \sigma) (-d\sigma) \\
&= \int_0^{t-t_{i-1}} \exp(A_C^i \sigma) \cdot B_C^i B_C^{iT} \cdot \exp(A_C^{iT} \sigma) d\sigma
\end{aligned} \tag{57}$$

From (56) we have

$$\dot{W}_i(t, t_{i-1}) = \exp[A_C^i(t - t_{i-1})] \cdot B_C^i B_C^{iT} \cdot \exp[A_C^{iT}(t - t_{i-1})] \tag{58}$$

From (57) we have

$$\dot{W}_i(t, t_{i-1}) = A_C^i \cdot W_i(t, t_{i-1}) + W_i(t, t_{i-1}) \cdot A_C^{iT} + B_C^i B_C^{iT} \tag{59}$$

Combining these two equation and evaluating at $t = t_i$, the following is found:

$$A_C^i W_i + W_i A_C^{iT} + B_C^i B_C^{iT} - \exp[A_C^i(t_i - t_{i-1})] B_C^i B_C^{iT} \exp[A_C^{iT}(t_i - t_{i-1})] = 0. \tag{60}$$

This equation has the form of a Lyapunov Matrix Equation: $AX + XA^T = -C$

where

$$X = W_i(t_i, t_{i-1})$$

$$A = A_C^i$$

$$C = B_C^i B_C^{iT} - \exp[A_C^i(t_i - t_{i-1})] B_C^i B_C^{iT} \exp[A_C^{iT}(t_i - t_{i-1})]$$

This equation can be solved for $W_i(t_i, t_{i-1})$ using some numeric method. In our case we used the Matlab LYAP command.

Using the previous results we can write an equation for the initial state $x_C^{i+1}(t_i)$ of the newly introduced controller as follows

$$x_C^{i+1}(t_i) = \exp[A_C^{i+1}(t_i - t_{i-1})] \cdot x_C^{i+1}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \exp[A_C^{i+1}(t_i - \tau)] \cdot B_C^{i+1} \cdot \hat{y}(\tau) \cdot d\tau \tag{61}$$

and by substituting (55) into (61) we have

$$x_C^{i+1}(t_i) = \exp[A_C^{i+1}(t_i - t_{i-1})] \cdot x_C^{i+1}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \exp[A_C^{i+1}(t_i - \tau)] \cdot B_C^{i+1} B_C^{i^T} \cdot \exp[A_C^i(t_i - \tau)] d\tau \cdot (62)$$

$$\cdot W_i^{-1}(t_i, t_{i-1}) \cdot [x_C^i(t_i) - \exp[A_C^{i+1}(t_i - t_{i-1})] \cdot x_C^i(t_{i-1})]$$

Next, define a new Grammian-like function as follows

$$W_{i+1,i}(t, t_{i-1}) = \int_{t_{i-1}}^t \exp[A_C^{i+1}(t - \tau)] \cdot B_C^{i+1} B_C^{i^T} \cdot \exp[A_C^i(t - \tau)] d\tau \quad (63)$$

Using the same logic we can show that this function can be found as a solution of a Sylvester Matrix Equation $AX + XB = -C$

where

$$A = A_C^{i-1}$$

$$B = A_C^{i^T}$$

$$C = B_C^{i+1} B_C^{i^T} - \exp[A_C^{i+1}(t_i - t_{i-1})] \cdot B_C^{i+1} B_C^{i^T} \cdot \exp[A_C^i(t_i - t_{i-1})]$$

Using $\hat{y}(t)$ to simulate the response of the $(i+1)^{st}$ controller on the interval $[t_{i-1}, t_i]$

initialized at $x_C^{i+1}(t_{i-1}) = 0$ leads to

$$x_C^{i+1}(t_i) = \int_{t_{i-1}}^{t_i} \exp[A_C^{i+1}(t_i - \tau)] B_C^{i+1} \hat{y}(\tau) d\tau \quad (64)$$

$$= W_{i+1,i}(t_i, t_{i-1}) W_i^{-1}(t_i, t_{i-1}) \times (x_C^i(t_i) - \exp[A_C^i(t_i - t_{i-1})] x_C^i(t_{i-1}))$$

The above formula indicates that an initial state for the $(i+1)^{st}$ controller, $x_C^{i+1}(t_i)$, that is in some sense close to the state that would have been reached had the controller been ‘warmed-up’ on the preceding time interval $[t_{i-1}, t_i]$, can be calculated from the controller

state-space data and states $x_C^i(t_{i-1})$ and $x_C^i(t_i)$ by solving two linear algebraic matrix equations.

2.6 Combining the D Method and Control Signal Interpolation

To analyze the influence of control signal interpolation on overall system properties we start with the controller architecture considered at the outset in Section 2.3, whose block diagram is presented in Figure 2.6

where $\tilde{u} = \begin{bmatrix} x_{C1} \\ z \\ y_2 \end{bmatrix}$

and

$$\begin{aligned} \tilde{A} &= A_{22}(\alpha), \\ \tilde{B} &= [A_{21}(\alpha) \quad B_{21}(\alpha) \quad B_{22}(\alpha)], \\ \tilde{C} &= C_2(\alpha), \\ \tilde{D} &= [C_1(\alpha) \quad D_1(\alpha) \quad 0]. \end{aligned}$$

Figure 2.6 – The original controller architecture

For each point design the interpolated portion of the controller is given by

$$\dot{x}_{C2}^i = A_{22}^i x_{C2} + \begin{bmatrix} A_{21}^i & B_{21}^i & B_{22}^i \end{bmatrix} \begin{bmatrix} x_{C1} \\ z \\ y_2 \end{bmatrix} \quad (65)$$

$$u^i = C_2^i x_{C2} + \begin{bmatrix} C_1^i & D_1^i & 0 \end{bmatrix} \begin{bmatrix} x_{C1} \\ z \\ y_2 \end{bmatrix}. \quad (66)$$

After applying the interpolation scheme given by (54) to (66) we have the following state representation of the corresponding portions of our interpolated controller

$$\begin{aligned} A_{21}(\alpha) &= \begin{bmatrix} A_{21}^i \\ A_{21}^{i+1} \end{bmatrix}, & A_{22}(\alpha) &= \begin{bmatrix} A_{22}^i & 0 \\ 0 & A_{22}^{i+1} \end{bmatrix}, \\ B_{21}(\alpha) &= \begin{bmatrix} B_{21}^i \\ B_{21}^{i+1} \end{bmatrix}, & B_{22}(\alpha) &= \begin{bmatrix} B_{22}^i \\ B_{22}^{i+1} \end{bmatrix}, \\ C_1(\alpha) &= \rho^i(\alpha)C_1^i + \rho^{i+1}(\alpha)C_1^{i+1}, & C_2(\alpha) &= \begin{bmatrix} \rho^i(\alpha)C_2^i & \rho^{i+1}(\alpha)C_2^{i+1} \end{bmatrix}, \\ D_1(\alpha) &= \rho^i(\alpha)D_1^i + \rho^{i+1}(\alpha)D_1^{i+1}, & D_2(\alpha) &= 0. \end{aligned} \quad (67)$$

Here we assume that Assumption A.1 is satisfied for the closed loop system consisting of the linearized plant and interpolated controller. Intuitively, this condition can be satisfied by taking linear point designs sufficiently close to each other. Actually, this intuition is valid only when each controller is itself stable and this may impose a possible restriction on the applicability of control signal interpolation. Assumption A.3 is also satisfied by this controller architecture because the controller family's entire $B_C(\alpha)$ matrix is constant implying the same for the $B_{22}(\alpha)$ submatrix.

Next we consider the modified controller structure that is designed according to the modified D method described in Section 2.4. The new controller architecture is presented in Figure 2.7.

Figure 2.7 – Modified controller architecture

where $\hat{u} = \begin{bmatrix} z \\ y_2 \end{bmatrix}$

$$\hat{A} = A_{22}(\alpha),$$

and $\hat{B} = \begin{bmatrix} A_{21}(\alpha) + A_{22}(\alpha)B_{21}(\alpha) & A_{22}(\alpha)B_{21}(\alpha) \end{bmatrix},$

$$\hat{C} = C_2(\alpha),$$

$$\hat{D} = \begin{bmatrix} C_1(\alpha) + C_2(\alpha)B_{21}(\alpha) & C_2(\alpha)B_{22}(\alpha) \end{bmatrix}.$$

so that Figure 2.7 corresponds with (45). The interpolated controller family is constructed by plugging (67) into (45) with a corresponding gain scheduled controller given by (51).

The design process for the gain scheduled controller that incorporates the D method and control signal interpolation can be summarized in the following four steps:

- Perform linear time invariant (LTI) point designs according to the controller architecture of Figure 2.6 for selected operating points;
- Perform control signal interpolation for the system in Figure 2.6 to get linear controller family (31) and verify closed-loop stability for all α ;
- Modify the design according to the D method to get the linear controller family (45) which satisfies the conditions of the Theorem 2.3;

- Implement the gain scheduled controller in the form (51) that satisfies the linearization requirements;

Chapter 3

THE BALL AND BEAM EXAMPLE

3.1 System Description

We apply the gain scheduling approach described in the preceding Chapter to the well-known ball and beam experiment whose mathematical model is taken from in [1] and draft is presented in Figure 3.1. The beam is rotated in a vertical plane by applying a torque at the center of rotation and the ball is free to roll. The beam angle is denoted by θ , ball position by r and applied torque by τ .

Figure 3.1 - The ball and beam system

The equations of motion are given by

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= B(x_1 x_4^2 - G \sin(x_3)) \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{-2Mx_1 x_2 x_4 - MGx_1 \cos(x_3) + u}{Mx_1^2 + J + J_b}
 \end{aligned} \tag{68}$$

where $B = M / (J_b / R^2 + M)$. System variables and their physical equivalents are summarized in a Table 3.1 and the remaining system parameters are listed in Table 3.2.

Table 3.1: System variables and their physical equivalents

System Variable	Physical equivalent	Units	Description
x_1	r	m	ball position
x_2	$\dot{r}$	m/s	ball velocity
x_3	θ	rad	beam angle
x_4	$\dot{\theta}$	rad/s	beam angular velocity
u	τ	N · m	applied torque

Table 3.2: Ball and Beam Parameters

Parameter	Description	Value
M	ball mass	0.05 kg
R	ball radius	0.01 m
J	beam inertia	0.02 kg m ²
J _b	ball inertia	2 · 10 ⁻⁶ kg m ²
G	Acceleration due to gravity	9.81 m/s ²

The regulated and measured outputs are taken to be

$$z = r_{ref} - x_1, \quad y = \begin{bmatrix} z \\ x_1 \end{bmatrix} \quad (69)$$

where r_{ref} denotes the reference ball position. A measured exogenous input is absent in this example.

In addition to specifying the regulated output and serving as an additional measurement for control, we select $x_1 = r$ as the scheduling variable. It follows that the plant's family of equilibria is characterized by

$$x^0(\alpha) = \begin{bmatrix} \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad u^0(\alpha) = MG\alpha, \quad r_{ref}^0(\alpha) = \alpha, \quad y^0(\alpha) = \begin{bmatrix} 0 \\ \alpha \end{bmatrix} \quad (70)$$

where we have imposed the constraint $x_3 = \theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ so that $\sin(x_3) = 0$ implies $x_3 = 0$.

The plant's family of linearizations is specified by

$$A(\alpha) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -BG & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{MG}{M\alpha^2 + J + J_b} & 0 & 0 & 0 \end{bmatrix}, \quad B_u(\alpha) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{M\alpha^2 + J + J_b} \end{bmatrix}, \quad (71)$$

$$C_z(\alpha) = [-1 \quad 0 \quad 0 \quad 0], \quad D_{zr}(\alpha) = 1, \quad C_y(\alpha) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_{yr}(\alpha) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

3.2 Controller Design

According to our control signal interpolation scheme, an interpolated linear controller family of the form (54) was constructed. For each point design we have the realization of the linear controller of the form (66). Here we used the “internal model principle” [19], [20] that requires a copy of reference signal dynamics in the controller structure to provide asymptotic tracking of this signal. Our controller was designed to track step reference inputs so our controller will contain an integral action. The elements of the controller are given by

$$\begin{aligned} A_{21}^i &= -B_u(\alpha_i)K_I^i, & A_{22}^i &= A(\alpha_i) - B_u(\alpha_i)K^i + K_e^i C_{y2}(\alpha_i) \\ B_{21}^i &= 0, & B_{22}^i &= K_e^i, & C_1^i &= -K_I^i, & C_2^i &= -K^i, & D_1^i &= 0. \end{aligned} \quad (72)$$

where K_I^i, K^i , and K_e^i are constructed to locate eigenvalues of

$$\begin{bmatrix} 0 & C_z(\alpha_i) \\ 0 & A(\alpha_i) \end{bmatrix} - \begin{bmatrix} 0 \\ B_u(\alpha_i) \end{bmatrix} \begin{bmatrix} K_I^i & K^i \end{bmatrix} \text{ and } A(\alpha_i) + K_e^i C_{y2}(\alpha_i)$$

at desired locations, so that, via the separation property,

$$\begin{bmatrix} A(\alpha_i) & B_u(\alpha_i)C_1^i & B_u(\alpha_i)C_2^i \\ C_z(\alpha_i) & 0 & 0 \\ B_{22}^i C_{y2}(\alpha_i) & A_{21}^i & A_{22}^i \end{bmatrix}$$

has the desired eigenvalues for each α_i . This implies Assumption A.1 is satisfied for $\alpha = \alpha_i$. It must also to be verified that Assumption A.1 is satisfied for all $\alpha \in \mathbf{O}$. This will be the case for design points that are sufficiently close to each other and linear controller that is itself stable.

Observer-based Type I point designs were constructed based on plant linearizations for $\{\alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\} = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ resulting in 6 linear point designs. To check the closed-loop system for hidden coupling terms we compare the closed-loop eigenvalues, closed-loop bode plots, and closed-loop step responses of the linearized closed-loop system and linear closed-loop system constructed of our interpolated controller and plant linearization for points inside of some working interval (α_i, α_{i+1}) . We chose $\alpha = 0.05$ m as the test point. To compute the closed loop linearization we used Matlab TRIM and LINMOD commands. The closed-loop eigenvalues of the linearized system are given by:

$$\begin{array}{ll} -13.3068 & \\ -22.4741 \pm 10.5697i & -9.8631 \pm 7.8140i \\ -4.5472 \pm 7.0548i & -2.6816 \pm 3.3378i \\ -3.2195 \pm 1.0912i & -4.4907 \pm 17.6817i \end{array}$$

The linear closed-loop system is given by

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} &= \begin{bmatrix} A + B_u D_C C_y & B_u C_C \\ B_C C_y & A_C \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + \begin{bmatrix} B_u D_C D_{yr} \\ B_C D_{yr} \end{bmatrix} r, \\ y &= \begin{bmatrix} C_z & 0 \end{bmatrix} \begin{bmatrix} x \\ x_c \end{bmatrix} + D_{zr} r \end{aligned} \tag{73}$$

For the linear of this form closed-loop eigenvalues are given by:

$$\begin{array}{ll} -13.3068 & \\ -22.4741 \pm 10.5697i & -9.8631 \pm 7.8140i \\ -4.5472 \pm 7.0548i & -2.6816 \pm 3.3378i \\ -3.2195 \pm 1.0912i & -4.4907 \pm 17.6817i \end{array}$$

The eigenvalues of the two systems match exactly.

To check the matching of closed-loop transfer functions, frequency and time domain analysis was performed. The difference in the unit step response of the two systems is given in Figure 3.2. Frequency response difference is presented in Figure 3.3. As can be seen from the plots, the step responses and Bode diagrams of the two systems are very close to each other. The small difference is caused by precision in a numeric calculation. So from this analysis and the eigenvalue comparison given above it can be concluded that both linearization requirements described at the start of Section 2.3 are satisfied. Thus, our controller has no hidden coupling terms as expected.

Figure 3.2 – Difference in tracking error response for a unit step input

Figure 3.3 – Difference in frequency response

3.3 Simulation Scheme

The testing and performance assessment of the design was conducted in Simulink. The simulation diagram for controller testing is presented in Figure 3.4. The diagram can be divided logically into five main blocks:

- reference input generation block that produces ramp-and-hold input to our system;
- controller block that consists of two controllers for upper and lower bound of the current working interval (α_i, α_{i+1}) ;
- interpolation block that interpolates the control signal based on the instantaneous value of scheduling variable;

- nonlinear plant block;
- control block that stops the simulation when the scheduling variable crosses upper or lower bound;

Figure 3.4 – Simulation Diagram

The nonlinear model of the ball and beam system is given by a Matlab system function (S-function) `NLPlant.m`, the full listing of which is given in Appendix A. For a specified design point linear controller calculations are performed using the Matlab function `control.m` which is given in Appendix B. Prior to starting the simulation we perform the initialization of all blocks in our simulation diagram using the Matlab script `init.m` which is given in Appendix C. In this script we calculate the linear observer-based

controllers for all design points using `control.m` and, based on initial ball displacement value, initialize the states of our Upper and Lower controllers, Integrator, Nonlinear plant and ramp generators.

The simulation is controlled from the Matlab script `run.m` presented in Appendix D. In this program the initialization script is called first and then start the simulation using the Matlab `SIM` command. As soon as the simulation is stopped because of a boundary crossing, the focus is returned to the main program which performs the controller switching according to the value of scheduling variable. Here the warm-up procedure described in detail in Section 2.5 is used. The simulation is restarted until the final simulation time is reached. Another task that is performed by this script is plotting the ball position and beam angle.

3.4 Performance Analysis

To analyze the performance of the closed-loop system with our interpolated controller, several nonlinear simulations were conducted. The state feedback and observer gain matrices were selected to place eigenvalues at

$$\left\{-5, -5 \cos\left(\frac{\pi}{8}\right) \pm j 5 \sin\left(\frac{\pi}{8}\right), -5 \cos\left(\frac{\pi}{4}\right) \pm j 5 \sin\left(\frac{\pi}{4}\right)\right\}$$

and

$$\left\{-10 \cos\left(\frac{\pi}{8}\right) \pm j 10 \sin\left(\frac{\pi}{8}\right), -10 \cos\left(\frac{\pi}{4}\right) \pm j 10 \sin\left(\frac{\pi}{4}\right)\right\}$$

respectively. Ramp-and-hold reference inputs were applied for the purpose of moving the

ball from an initial position of -0.45 m to a final position of 0.45 m. Figure 3.5 illustrates the necessity of a gain scheduled controller for the Ball and Beam system. As it can be seen from the plot, the linear controller designed for -0.45 m makes the closed-loop system unstable. To find the limits of the proposed controller architecture, a series of simulations with increasing slope of the reference ramp-and-hold signal was performed. Figures 3.6 through 3.17 show the ball position and beam angle responses for reference input slopes of $0.5, 1.0, 1.5, 2.0, 2.2$ and 2.5 m/s. As is typically the case when scheduling linear time-invariant controllers designed about a family of equilibria, the performance of the nonlinear closed-loop system agrees with that predicted by linear analysis of the family of closed-loop linearizations as long as all exogenous signals are sufficiently slowly-varying. Here, the time rate-of-change of the lone exogenous input $r_{ref}(t)$ corresponds to the ramp slope and we see that performance degrades as this slope increases.

In addition, to analyze the system behavior at controller switching times, we studied the transient responses at two different boundaries: -0.4 m and 0.3 m. Corresponding magnified transient responses are given in Figures 3.18 and 3.19. Transient effects due to new controllers being brought on-line at these switching times are negligible, indicating the effectiveness of our controller initialization scheme described in Section 2.5. Thus, our assumption that $x_C^{i+1}(t_{i-1}) = 0$ does not introduce any undesired transient effects.

Figure 3.5 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 1.5 m/s: linear controller

Figure 3.6 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 0.5 m/s.

Figure 3.7 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 0.5 m/s.

Figure 3.8 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 1.0 m/s.

Figure 3.9 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 1.0 m/s.

Figure 3.10 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 1.5 m/s.

Figure 3.11 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 1.5 m/s.

Figure 3.12 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 2.0 m/s.

Figure 3.13 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.0 m/s.

Figure 3.14 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 2.2 m/s.

Figure 3.15 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.2 m/s.

Figure 3.16 - Ball position $r(t)$ and reference ball position $r_{ref}(t)$ versus time for ramp-and-hold slope of 2.5 m/s.

Figure 3.17 - Beam angle $\theta(t)$ versus time for ramp-and-hold slope of 2.5 m/s.

Figure 3.18 - Transient response for controller switch from -0.5 m to -0.4 for ramp-and-hold with slope of 1.5 m/s

Figure 3.19 - Transient response for controller switch from 0.2 m to 0.3 for ramp-and-hold with slope of 1.5 m/s

Figure 3.20 shows the magnified portion of the transient response for the controller switch from 0 m to 0.1 m for the slope of the reference input signal equal to 2.2 m/s. As it can be seen from the plot there is a significant transient response at this time, but we conjecture that this is not caused by controller switch. To prove this, we compare our controller to a controller that is similar in several respects, but does not involve the controller switching. This alternate controller also utilizes a Type I observer-based controller structure except the controller gains are calculated analytically for all values of the scheduling variable [21]. Figures 3.21 and 3.22 show the transient responses of both controllers. As it can be seen from the plots, the gain scheduled controller using control signal interpolation provides similar performance as compared with the controller

with analytically scheduled gains. This similarity supports the conclusion that our controller initialization scheme satisfactory eliminates transient effects associated with controller switching.

The achieved results are sufficiently better when the results presented in [22]. The improvement in performance is caused by a better choice of closed-loop eigenvalues at the observer-based type I controller design step. The final eigenvalue placement was performed based on the experiments.

Figure 3.20 - Transient response for controller switch from 0 m to 0.1 m for ramp-and-hold with slope of 2.2 m/s

Figure 3.21 – Proposed controller vs. controller with analytical gain scheduling: Ball Position for ramp-and-hold with slope of 1.5 m/s

Figure 3.22 – Proposed controller vs. controller with analytical gain scheduling:
Beam Angle for ramp-and-hold with slope of 1.5 m/s

Chapter 4

CONCLUDING REMARKS

Gain scheduling based on plant linearizations about equilibria implicitly imposes a linearization requirement on the controller. Existence conditions for gain scheduled controllers possessing the required linearization property are quite restrictive because they involve partial differential equations.

In practice, gain scheduled controllers are often implemented as LPV controllers based on interpolated linear point designs. Controllers designed in this way will typically fail to satisfy a basic linearization requirement resulting in the presence of hidden coupling terms that may adversely affect performance.

In this thesis, an approach combining a modified D method with control signal interpolation that leads to a gain scheduled controller that retains an LPV structure but is free of hidden coupling terms was considered. Further, the need for time-derivatives of certain controller inputs is avoided here. This design methodology has successfully been applied to the ball and beam experiment. It has been shown via frequency domain and time domain analysis that the designed gain scheduled controller has no hidden coupling terms. Also, an original algorithm for controller warm up has been developed that

provides transient free switching of controllers. Nonlinear simulations indicate acceptable performance subject to the limitations inherent in linearization scheduling.

Future research may be concentrated on applying the approach on a more complex system with several scheduling variables. This will necessitate a different switching scheme. Also it is interesting to check the performance of this method on a physical apparatus, which is not difficult because the ball and beam experiment is frequently used as an example problem in control theory classes at universities. This work can also be applied to hardware platforms currently available at Ohio University such as the Brumby UAV or Quanser Consulting's "UFO". Another thing that can improve the performance of the controller is a more complex "internal model" which will allow tracking of different exogenous signals with no steady-state error.

Future work is also required on the analysis of our warm-up scheme and the investigation of alternative schemes. For example, when point designs have the same order, the state of the controller that is being retired might be used as the initial state for the new controller that is coming on-line. Another interesting direction for future research is controller failure prediction. Existing stability results only provide a qualitative sufficient condition. A more detailed analysis is required to derive less conservative and more accurate stability indicators.

REFERENCES

- [1] Lawrence D.A. and W. J. Rugh. "Gain scheduling dynamic linear controllers for a nonlinear plant". *Automatica*, **31**(3), 381-390, 1995.
- [2] Kelemen, M. "A stability property." *IEEE Transactions on Automatic Control*, **31**(8), 766-768, 1986.
- [3] Khalil, H.K., Kokotovic, P.V. "On stability properties of nonlinear systems with slowly varying inputs." *IEEE Transactions on Automatic Control*, **36**(2), 229, 1991.
- [4] Lawrence, D.A., Rugh, W.J. "On a stability theorem for nonlinear systems with slowly varying inputs." *IEEE Transactions on Automatic Control*, **35**(7), 860-864, 1990.
- [5] Kaminer, I., A. M. Pascoal, P. P. Khargonekar, and C. Thomson. "A velocity algorithm for the implementation of gain scheduled controllers." *Automatica*, **31**(8), 1185-1191, 1995.
- [6] Kaminer, I., A. M. Pascoal, P. P. Khargonekar, and E. Coleman. "A velocity algorithm for the implementation of nonlinear gain scheduled controllers." Internal Report, Naval Postgraduate School, 1994.
- [7] W.J. Rugh and J.S. Shamma. "Research on gain scheduling." *Automatica*, **36**(10), 1401-1425, 2000.
- [8] Shamma, J.C. & Cloutier, J.R. "Trajectory scheduled missile autopilot design." *Proceedings of the first IEEE conference on control applications*, Dayton, OH, September, 1992.
- [9] Sureshbabu, N., & Rugh, W.J. "Output regulation with derivative information." *IEEE Transactions on Automatic Control*, **40**(10), 1755-1766, 1995.
- [10] Tan, H., & Rugh, W.J. "Nonlinear overtaking optimal control: Sufficiency, stability, and approximation." *IEEE Transactions on Automatic control*, **43**(12), 1703-1718, 1998.
- [11] Leonessa, A., Haddad, W.M., & Chellaboina, V. "Nonlinear system stabilization via stability-based switching." *Proceedings of the 37th IEEE conference on decision and control*, Tampa, FL, 2983-2996, 1998.

- [12] Shamma, J.S., & Xiong, D. "Set-valued methods for linear parameter varying systems." *Automatica*, **35**(6), 1081-1089, 1999.
- [13] Tu, K.-H., & Shamma, J.S. "Nonlinear gain-scheduled control design using set-valued methods." *Proceedings of the American control conference*, Philadelphia, PA, 1998.
- [14] Bamieh, B., & Giarré, L. "Identification of linear parameter varying models." *Proceedings of the 38th IEEE conference on decision and control*, Phoenix, AZ, December 1999.
- [15] Wu, F., Grigoriadis, K.M. "LPV-based control of systems with amplitude and rate actuator saturation constraints." *Proceedings of the American control conference*, San Diego, CA, 3191-3195, June 1999.
- [16] Lawrence, D.A., Kelly J.H., and Evers J.H. "Gain scheduled missile autopilot design using a control signal interpolation technique." *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, AIAA Paper No. 98-4418, Boston, MA, 1998.
- [17] Kelly, J.H., Evers, J.H, Korn, R.A, Lawrence, D.A. "Scheduling dynamic compensators via control signal interpolation." *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, AIAA Paper No. 99-3974, Portland, OR, 1999.
- [18] Hauser, J., Sastry S., and Kokotovic P.V. "Nonlinear control via approximate input-output linearization: the ball and beam example." *IEEE Transactions on Automatic Control*, **37**(3), 392-398, 1992.
- [19] Ogata K. **Modern Control Engineering**. Prentice-Hall, 824-829, 1997.
- [20] Dorf R.C., Bishop R.H. **Modern Control Systems.- 8th ed.**, Addison-Wesley, 662-665, 1998.
- [21] Lawrence, D.A. "Gain scheduled controllers with guaranteed linearization properties." *Proceedings of the 2001 American Control Conference*, pp. 4128-4133, Arlington, Virginia, June 2001.
- [22] Ignatov R.P. and Lawrence D.A. "Gain Scheduling via Control Signal Interpolation: The Ball and Beam Example." *Proceedings of the 33rd Southeastern Symposium on System theory*, Ohio University, Athens, Ohio, 419-423, March 2001.

APPENDIX A: Matlab plant model (NLPlant.m)

```

Matlab plant model (ruslan.m)
function [sys,x0,str,ts] = ruslan(t,x,u,flag,X0)
%Nonlinear plant
%
switch flag,

% Initialization %
case 0,
[sys,x0,str,ts] = mdlInitializeSizes(X0);

% Derivatives %
case 1,
sys = mdlDerivatives(t,x,u);

case 2,
sys=mdlTerminate(t,x,u);

% Outputs %
case 3,
sys=mdlOutputs(t,x,u);
% Terminate %
case 9,
sys=mdlTerminate(t,x,u);

% Unexpected flags %
otherwise
error(['Unhandled flag = ',num2str(flag)]);
end

%
function [sys,x0,str,ts]=mdlInitializeSizes(X0)

sizes = simsizes;
sizes.NumContStates = 4;
sizes.NumDiscStates = 0;

```

```

sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1; % at least one sample time is needed
sys = simsizes(sizes);

% initialize the initial conditions
x0=X0;
%
str = [];
ts = [0 0]; % continuous sample time: [period, offset]
%
function sys = mdlDerivatives(t,x,u)
B=0.7143;
G=9.81;
M=0.05;
J=0.02;
Jb=0.000002;
sys(1) = x(2);
sys(2)=B*(x(1)* x(4)*x(4)-G*sin(x(3)));
sys(3) = x(4);
sys(4)=(1/(M*x(1)*x(1)+J+Jb))*(u-2*M*x(1)*x(2)*x(4)-M*G*x(1)*cos(x(3)));
%
function sys=mdlOutputs(t,x,u)
sys = x(1);
% end mdlOutputs
%
function sys=mdlTerminate(t,x,u)
sys = [];
% end mdlTerminate

```

APPENDIX B: Linear controller calculation (control.m)

```

Linear controller calculation (control.m)
function con=control(alpha)
%symbolic variables declaration
x1=sym('x1');
x2=sym('x2');
x3=sym('x3');
x4=sym('x4');
tau=sym('tau');
B=sym('B');
G=sym('G');
M=sym('M');
J=sym('J');
Jb=sym('Jb');
%

%Plant linealirezation
tmp1=x2;
tmp2=B*(x1*x4^2-G*sin(x3));
tmp3=x4;
tmp4=(tau-(2*M*x1*x2*x4)-(M*G*x1*cos(x3)))/(M*x1^2+J+Jb);
AAA=[tmp1;tmp2;tmp3;tmp4];
AAA=simple(AAA);
X=[x1 x2 x3 x4];
AA=jacobian(AAA,X);
AA=simple(AA);
tmpB1=diff(tmp1,tau);
tmpB2=diff(tmp2,tau);
tmpB3=diff(tmp3,tau);
tmpB4=diff(tmp4,tau);
BB=[tmpB1;tmpB2;tmpB3;tmpB4];
C=[1,0,0,0];
D=[0 1 0 0 0];
%

%Make substitutions
x1=alpha;%set the schedulling variable "r" = x1

```

```

%
x2=0;
x3=0;
x4=0;
B=0.7143;
G=9.81;
M=0.05;
J=0.02;
Jb=0.000002;
tau=M*G*x1;

A=subs(AA);
B=subs(BB);
A=double(A);
B=double(B);
A11=A;
B11=B;
C11=C;
D11=D;
% end of substitution

%Pole placement controller design-internal model for step input
A1=[-C;A];
z=[0;0;0;0;0];
A1=[z A1];
B1=[0;B];
D1=0;
C1=[0 C];
p=[-5 -5*cos(pi/8)+j*5*sin(pi/8) -5*cos(pi/8)-j*5*sin(pi/8) -5*cos(pi/4)+j*5*sin(pi/4) -
5*cos(pi/4)-j*5*sin(pi/4) ];
K=place(A1,B1,p);
K2345=[K(2) K(3) K(4) K(5)];
K1=K2345;
K11=K(1);

%Observer-controller design
%r=x1=0

```

```

Pe=[-10*cos(pi/8)+j*10*sin(pi/8) -10*cos(pi/8)-j*10*sin(pi/8) -
10*cos(pi/4)+j*10*sin(pi/4) -10*cos(pi/4)-j*10*sin(pi/4)];
Ke=place(A',C',Pe);
Ke1=Ke;
%Linear controller #1: integrator moved to the output of controller

Alc1=[A11-B11*K1-Ke1*C11];
Blc1=[-B11*KI1 (A11-B11*K1-Ke1*C11)*Ke1 ];
Clc1=[-K1];
Dlc1=[-KI1 -K1*Ke1];
con=ss(Alc1,Blc1,Clc1,Dlc1);

```


APPENDIX C: System initialization (init.m)

```

clc
clear

alpha=0:0.1:0.5;
k=0;
for i=1:6
    GSC(:,i)=control(alpha(i));
end
G=9.81;
M=0.05;
%Initial displacement
x1=-0.45;
%
%Initial condition for the Integrator
Integ=M*G*x1;
%
%Initial condition for the nonlinear plant
X0=[x1; 0;0;0]
%
if (abs(x1)>=0)&(abs(x1)<0.1)
    k=1;
elseif(abs(x1)>=0.1)&(abs(x1)<0.2)
    k=2;
elseif (abs(x1)>=0.2)&(abs(x1)<0.3)
    k=3;
elseif (abs(x1)>=0.3)&(abs(x1)<0.4)
    k=4;
elseif (abs(x1)>=0.4)&(abs(x1)<0.5)
    k=5;
%elseif (abs(x1)>=0.4)&(abs(x1)<0.5)
% k=6;
end
%Lower controller
AlcL=GSC(:,k).a;
BlcL=GSC(:,k).b;

```

```

ClcL=GSC(:,k).c;
DlcL=GSC(:,k).d;
%Upper controller
AlcU=GSC(:,k+1).a;
BlcU=GSC(:,k+1).b;
ClcU=GSC(:,k+1).c;
DlcU=GSC(:,k+1).d;
%Initial conditions for a lower and upper controller
X0L=-1*inv(AlcL)*BlcL*[0;x1];
X0U=-1*inv(AlcU)*BlcU*[0;x1];
%Boundary conditions
L=(k-1)*0.1;
U= k*0.1;
Lc=L-eps;
Uc=U+eps;
%Limits for interpolator
D=[Lc;Uc];
%
number=k;

```

APPENDIX D: Simulation procedure (run.m)

```

clear
clc
%Initialization of controllers and setting up initial conditions
init;
%
tsim=[];
xsim=[];
lastt=0;
tstop=10.0;
step=0;
start=[];
y=[];
while lastt<tstop
%start of simulation
sim('ball_obs4',[lastt tstop]);
y=[y; rmp];
%
tsim=[tsim;tout];
start=length(tsim)-length(tout)+1;
lastt=tout(length(tout))
xsim=[xsim;xout];
sz=size(xsim);
%Initial condition for the Integrator
Integ=xsim(sz(1),1);
%
%Noinlinear plant initialization
X0=[xsim(sz(1),2) xsim(sz(1),3) xsim(sz(1),4) xsim(sz(1),5)];
%
%Staying controller determination
chk=xsim(sz(1),2);
if (abs(chk)>=U)
%Initial condition for the staying controller
Stay=[xsim(sz(1),10); xsim(sz(1),11); xsim(sz(1),12); xsim(sz(1),13)];
%
%Initialization of the new pair of controllers

```

```

number=number+1
%new upper controller
AlcU=GSC(:, :, number+1).a;
BlcU=GSC(:, :, number+1).b;
ClcU=GSC(:, :, number+1).c;
DlcU=GSC(:, :, number+1).d;
%
%Second controller stays as lower
AlcL=GSC(:, :, number).a;
BlcL=GSC(:, :, number).b;
ClcL=GSC(:, :, number).c;
DlcL=GSC(:, :, number).d;
%
%Initial condition of the staying controller at the beginning of current simulation step
X1T0=[xsim(start,10); xsim(start,11); xsim(start,12); xsim(start,13)];
%
%Introducing controller state calculation
Clyap=BlcL*BlcL'-
expm(AlcL*tout(length(tout)))*BlcL*BlcL'*expm(AlcL'*tout(length(tout)));
WR1=lyap(AlcL,Clyap);
Csylv=BlcU*BlcL'-
expm(AlcU*tout(length(tout)))*BlcU*BlcL'*expm(AlcL'*tout(length(tout)));
W2=lyap(AlcU,AlcL',Csylv);
X0U=W2*inv(WR1)*(Stay-expm(AlcL*tout(length(tout)))*X1T0);
%
%Staying controller state
X0L=Stay;
%
%Update of limits
U=U+0.1;
L=L+0.1;
Lc=L-eps;
Uc=U+eps;
D=[Lc Uc];
%
end%if

```

```

if (abs(xsim(sz(1),2))<=L)
%if (abs(chk)<=L)

%Initial condition for the staying controller
Stay=[xsim(sz(1),6); xsim(sz(1),7); xsim(sz(1),8); xsim(sz(1),9)];
%
%Initialization of the new pair of controllers
number=number-1
%new Lower controller
AlcL=GSC(:, :, number).a;
BlcL=GSC(:, :, number).b;
ClcL=GSC(:, :, number).c;
DlcL=GSC(:, :, number).d;
%
%Second controller stays as Upper
AlcU=GSC(:, :, number+1).a;
BlcU=GSC(:, :, number+1).b;
ClcU=GSC(:, :, number+1).c;
DlcU=GSC(:, :, number+1).d;
%
%Initial condition of the staying controller at the beginning of current simulation step
X1T0=[xsim(start,6); xsim(start,7); xsim(start,8); xsim(start,9)];
%X1T0=[xsim(start,10); xsim(start,11); xsim(start,12); xsim(start,13)];
%
%Introducing controller state calculation
Clyap=BlcU*BlcU'-
expm(AlcU*tout(length(tout)))*BlcU*BlcU'*expm(AlcU'*tout(length(tout)));
WR1=lyap(AlcU,Clyap);
Csylv=BlcL*BlcU'-
expm(AlcL*tout(length(tout)))*BlcL*BlcU'*expm(AlcU'*tout(length(tout)));
W2=lyap(AlcL,AlcU',Csylv);
X0L=W2*inv(WR1)*(Stay-expm(AlcU*tout(length(tout)))*X1T0);
%
%Staying controller state
X0U=Stay;
%
%Update of limits

```

```
U=U-0.1
L=L-0.1
abs(xsim(sz(1),2))
Lc=L-eps;
Uc=U+eps;
D=[Lc Uc];
end%if

end%while

figure(1)
plot(tsim,xsim(:,2),tsim,y);
grid
xlabel('Time, [sec]')
ylabel('Ball position, [m]')

figure(2)
plot(tsim,xsim(:,4)*180/pi);
grid
xlabel('Time, [sec]')
ylabel('Beam angle, [Degrees]')
```