DEVELOPMENT OF A ROBUST FRAMEWORK FOR CONTROLLING HIGH PERFORMANCE TURBOFAN ENGINES

ROBERT MIKLOSOVIC

Bachelor of Electrical Engineering

Cleveland State University

June, 1996

Master of Science in Electrical Engineering

Cleveland State University

August, 2001

submitted in partial fulfillment of requirements for the degree

DOCTOR OF ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

May, 2006

This dissertation has been approved
for the Department of Electrical and Computer Engineering
and the College of Graduate Studies by

____

Dissertation Committee Chairperson, Zhiqiang Gao

____

Department/Date

____

Dan Simon

____

Department/Date

____

Paul Lin

____

Department/Date

____

Majid Rashidi

____

Department/Date

____

Pratibha Ghatage

____

Department/Date

To my wonderful wife Tina

ACKNOWLEDGEMENTS

I would like to give special thanks to Dr. Zhiqiang Gao for his endless patience and guidance over the past ten years. I also appreciate Dr. Dan Simon, Dr. Paul Lin, Dr. Majid Rashidi, and Dr. Pratibha Ghatage for taking time out of their schedules and agreeing to serve on my committee. I am also thankful to my peers for sharing their ideas freely and for their opinions on the details, especially Aaron Radke and Brian Fast, but also including Qing Zheng, Frank Goforth, Wankun Zhou, Zhan Ping, and Bosheng Sun.

Thanks to Dr. Sanjay Garg for the opportunity at NASA Glenn Research Center and support under grant NGT3-52387 as well as to Jonathan Litt and Khary Parker for sharing their knowledge and time with me. I would like to express my appreciation to Ken Dearborn and Rich Brown at Dearborn Inc. for their continuous support over the past eighteen years.

My deepest gratitude goes to my wife Tina for believing in me and for being right by my side at all times. I would also like to thank Pastor Steve Carmany and of all my friends at the United Church of Huntington for their fellowship, but most of all I thank God for the blessings bestowed upon me.

DEVELOPMENT OF A ROBUST FRAMEWORK FOR CONTROLLING HIGH PERFORMANCE TURBOFAN ENGINES

ROBERT MIKLOSOVIC

ABSTRACT

This research involves the development of a robust framework for controlling complex and uncertain multivariable systems. Where mathematical modeling is often tedious or inaccurate, the new method uses an extended state observer (ESO) to estimate and cancel dynamic information in real time and dynamically decouple the system. As a result, controller design and tuning become transparent as the number of required model parameters is reduced. Much research has been devoted towards the application of modern multivariable control techniques on aircraft engines. However, few, if any, have been implemented on an operational aircraft, partially due to the difficulty in tuning the controller for satisfactory performance. The new technique is applied to a modern two-spool, high-pressure ratio, low-bypass turbofan with mixed-flow afterburning. A realistic Modular Aero-Propulsion System Simulation (MAPSS) package, developed by NASA, is used to demonstrate the new design process and compare its performance with that of a supplied nominal controller. This approach is expected to reduce gain scheduling over the full operating envelope of the engine and allow a controller to be tuned for engine-to-engine variations.

TABLE OF CONTENTS

Page

NOMENCLATURE

LPC:	Low pressure compressor (Booster tip)
HPC:	High pressure compressor (Booster hub)
LPT:	Low pressure turbine
HPT:	High pressure turbine
VABI:	Variable-area bypass injectors
MAPSS:	Modular aero-propulsion system simulation
GUI:	Graphic user interface
CLM:	Component level model

Operating Point

pla:	Power lever angle
alt:	Altitude
xm:	Mach number

CLM Actuator Inputs

wf36:	Main burner fuel flow
a8:	Variable nozzle exit area
a16:	Rear bypass door variable area
a14:	Forward bypass door variable area

stp2:	Fan inlet guide vane angle
stp27:	HPC variable stator vane angle
stp27d:	Booster inlet guide vane angle

CLM Sensor Outputs

t3:	HPC exit temperature
ps3:	HPC exit static pressure
xn2:	Fan rotor speed
xn25:	Core rotor speed
fn:	Net thrust
sm2:	Fan stall margin
pcn2r:	Percent corrected fan speed
t56:	LPT exit temperature
p16qp56:	Mixer pressure ratio
ps56:	LPT exit static pressure @ mixer
p27:	HPC inlet pressure
t27:	HPC inlet temperature

Unmeasured Variables

w3:	HPC exit mass flow
t4:	Burner exit total temperature

Engine States

xn2:	Fan rotor speed
xn25:	Core rotor speed
tmpc:	Average metal temperature across the burner

Performance Parameters

lepr:	Liner engine pressure ratio
eprs:	Engine pressure ratio
cepr:	Core engine pressure ratio
etr:	Engine temperature ratio

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

INTRODUCTION

The jet engine, also known as the gas turbine engine, originated from many different sources. As far back in history as 250 B.C., Heron of Alexandria was said to have built the first reaction engine in a closed sphere with pressurized steam [1]. As illustrated in Figure 1, the steam escaped through two tubes which acted as jet nozzles. This force caused the sphere to rotate on its axis. Rockets made their way into existence in China between the eleventh and thirteenth century A.D. They discovered that a firecracker open at one end would act as a jet nozzle, and when tied to an arrow, it would dart in a straight line.

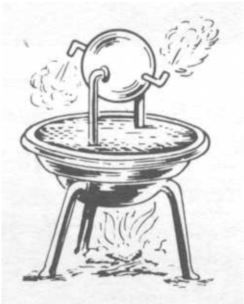


Figure 1: Heron's Reaction Engine

From the mid-1800s to the early 1900s, different engines and airplanes were invented, constructed, and taken into flight. In 1908 Rene Lorin made a revolutionary proposal to compress air with a piston and mix it with fuel. It would then be burned, producing "pulses of hot gases that would be expelled through a nozzle to generate propelling power" [1]. Between the years of 1929 and 1941, two men separately developed engines which would then be used in successful jet-powered aircraft during World War II. Sir Frank Whittle of Great Britain and Han von Ohain of Germany both built and flew their engines: von Ohain's first flight was in 1939 and Whittle's was in 1941 [2].

Since then, jet engine technology has advanced to the point of being considered one of the most complex systems in existence. Chapter II gives a brief introduction to the different types of jet engines and the progression of their mathematical models. A simulation package of a modern low bypass turbofan with a fully functional digital controller was developed in cooperation with NASA. Part of this research included completely rewriting its interface to automate the testing of new controllers. This amounted to over 3000 new lines of code. Functionality was also added to create, simulate, and compare of 27 different types of linear models proposed in [3]. Details of the supplied multi-mode digital controller are then given. The primary objective of control is to maintain a level performance within safe operating limits of the engine.

Improvements to control algorithms promise a high return on investment with little or no hardware modification. Chapter III outlines several model based control methods that have been proposed, yet none have ever been implemented because of the inability to adjust the controller for peak performance on an actual aircraft. Multivariable proportional-integral-derivative (PID) controllers are currently used in practice, but they require the complex adjustment of several parameters. This manual adjustment of controller parameters is referred to as tuning. A deeper look at current control structures for complex multivariable systems in general uncovers a few inherent problems. As a result, several new disturbance rejection and error based control methods are investigated. A technique referred to as Active Disturbance Rejection Control (ADRC) appears to be a practical solution because it provides high performance through a simple design procedure.

Chapter IV proposes a new framework for controlling complex and uncertain systems. A discrete implementation and generalization of the extended state observer (ESO) is first proposed to improve the performance of ADRC. Based on ADRC concepts, a new two degree-of-freedom (DOF) control structure, referred to as generalized PID (GPID), is then proposed. Finally, both ADRC and GPID are extended to control multiple-input multiple-output (MIMO) systems of arbitrary order. A gain and order estimation method is also proposed.

Multivariable ADRC is applied to the jet engine in Chapter V. The new algorithm is simulated at a few operating points and the results are discussed. Concluding remarks, given in Chapter VI, include future research directions as a direct result of this work.

CHAPTER II

PROBLEM FORMULATION

This chapter begins with a brief introduction of jet engines, focusing on the low bypass turbofan. Different models are then presented, followed by a discussion of control objectives, performance specifications, and engine limits. The details of the nominal engine controller are also given.

2.1 Types of Jet Engines

Before there were jet engines on aircraft, a piston engine was used to turn a propeller. The propeller would produce a small acceleration on a relatively large mass of air. Conversely, the gas turbine engine produces a relatively large acceleration directly on a small mass of air [1]. This design allows for quieter operation, better efficiency, and produces much more thrust for a given engine weight. The simplest gas turbine engine is the turbojet. The cylindrical engine case houses an axial compressor, burner, and turbine. The ambient air entering the engine is first compressed by the multi-stage compressor, mixed with fuel, and combusted in the burner section to produce thrust. Roughly seventy-five percent of the thrust produced is consumed by the multi-stage turbine, which rotates the compressor through a common shaft [1]. The compressor-shaft-turbine combination is called a spool. The remaining twenty-five percent of the thrust exits through the rear nozzle to propel the aircraft.

There are many variations to the basic turbojet design, formed by combining the concepts of the twin-spool turbojet, turboprop, high and low bypass turbofan, and afterburner concepts. When an engine is comprised of two spools, the incoming air passes through a low pressure compressor (LPC) and a high pressure compressor (HPC) before it is mixed with fuel and combusted. The thrust produced is absorbed first by a low pressure turbine (LPT), and then by a high pressure turbine (HPT) before the remaining thrust exits through the rear nozzle. The dual-rotor design improves compressibility mismatch across the compressor, the key component affecting performance [2]. The LPC and HPC rotors, functioning as a unit, greatly improve efficiency at off-design operating points while achieving higher pressure ratios. The design of variable inlet stator guide vanes has also significantly improves compressor performance. The combination of the HPC, combustor, and HPT is commonly referred to as the engine core. The turboprop has a propeller that is connected to the low pressure spool shaft through a gearbox.

The turbofan engine is a jet engine with a bypass stream. In this configuration, a large fan is diametrically attached to the front compressor stage or stages allowing a portion of the air to bypass the engine core and directly produce thrust. Although similar in concept to the turboprop, the bypass duct is designed so that the flow passing through it is not affected by the vehicle's airspeed. The ratio between the secondary bypass air stream and the primary air stream flowing through the engine core is called the bypass ratio. In general there are two basic types of turbofan engines; high-bypass turbofan engines, shown in Figure 2 which are typical of commercial and transport aircraft, and low-bypass military jet engines.

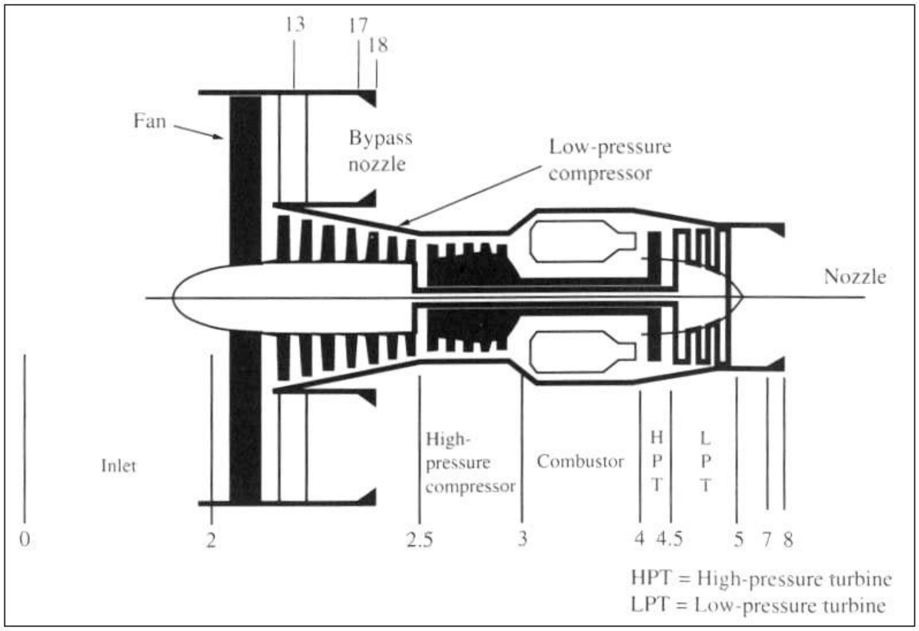


Figure 2: Schematic of a High-Bypass Turbofan

The afterburner, characteristic of military or high speed aircraft, is a long cylindrical section positioned between the rear turbines and the exhaust nozzle of the jet engine. Here the exhaust gases are mixed with fuel and re-combusted to thoroughly burn any un-combusted products. When implemented on a turbofan, it is not uncommon for the bypass stream to be mixed with the core stream through variable area bypass injectors (VABI) before entering the afterburner. This is referred to as mixed-flow afterburning. The exhaust nozzles have variable geometry to minimize the backlash of the reheat operation [4]. A schematic is shown in Figure 3 with labeled actuators (top) and sensors (bottom) [5].

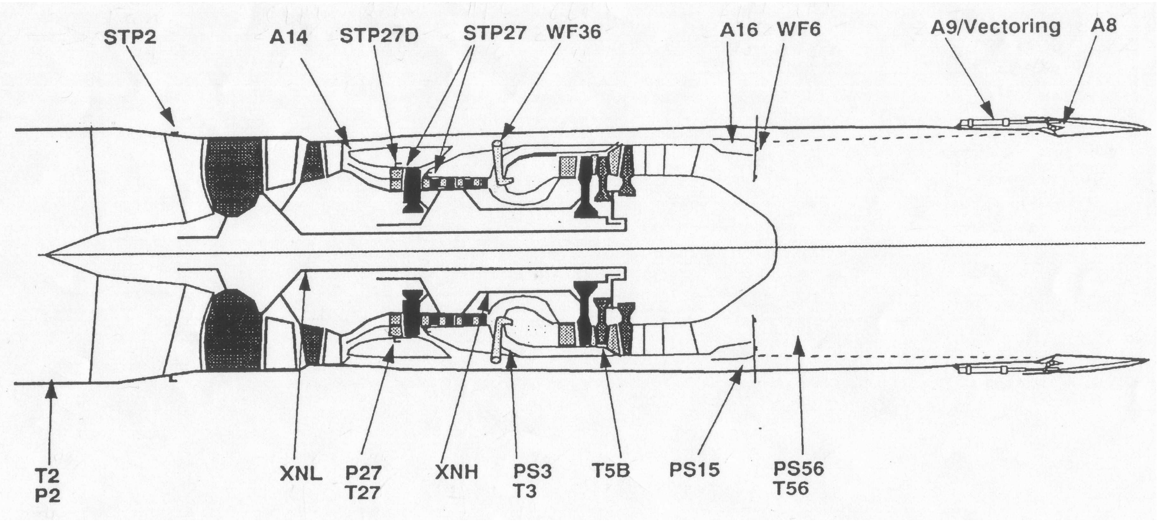


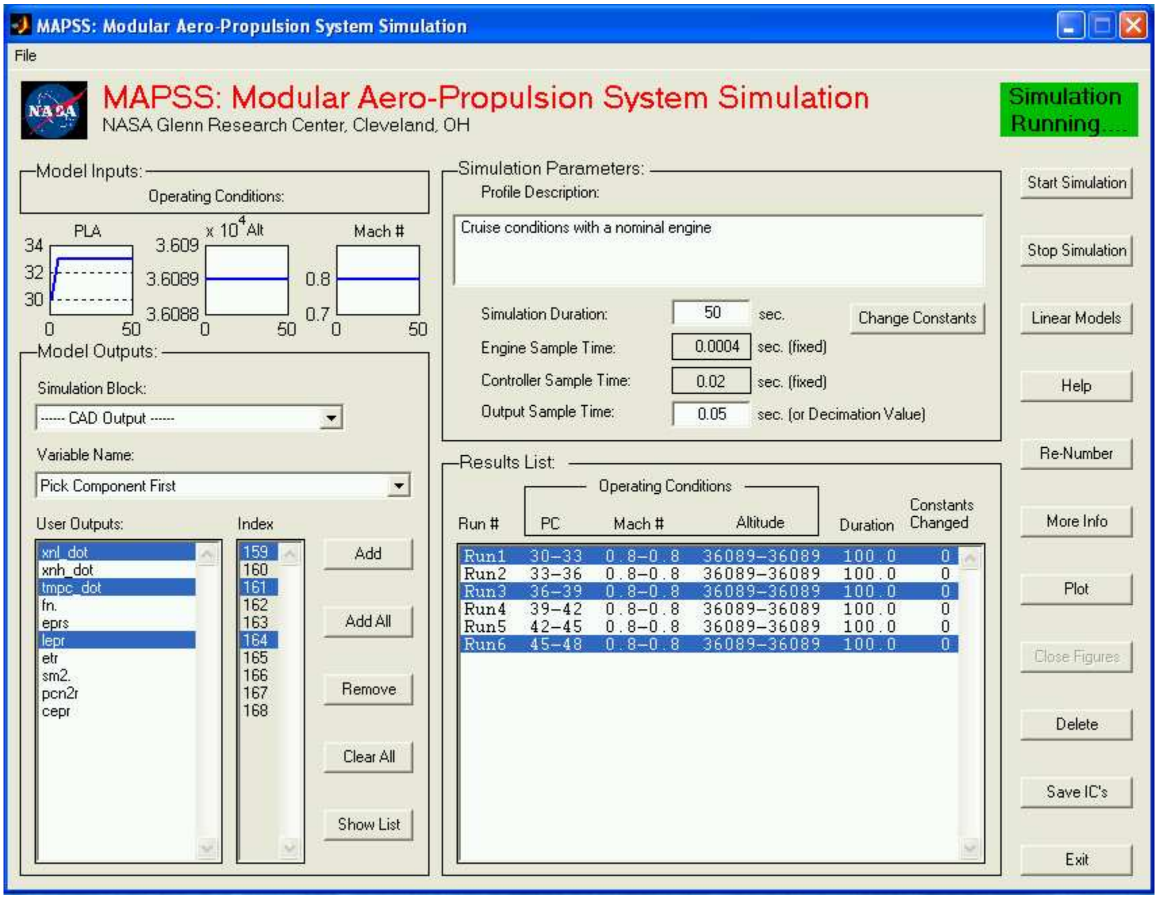
Figure 3: Schematic of a Low-Bypass Turbofan with Afterburner

2.2 Modular Aero-Propulsion System Simulation

Recently, the NASA Glenn Research Center has developed a Modular Aero-Propulsion System Simulation (MAPSS) package in Simulink® that incorporates a working military jet engine and a digital controller [6]. Both are completely configurable through a Graphic User Interface (GUI), allowing easy access to any signal in the engine or controller. This non real time multi-rate package is freely-distributed and will serve as a basis for simulation and controller design.

The engine used in MAPSS is a two-spool, high pressure ratio, low bypass turbofan with mixed-flow afterburning, similar to that in Figure 3. A multi-stage fan provides initial compression followed by a multi-stage HPC. An annular combustor, situated between the compressor and turbine stages, increases the energy state of the core gases to provide adequate fan and compressor energy requirements as well as sufficient thrust for the aircraft [7]. These gases are then expanded through high and low pressure turbines which drive the compressor and fan, respectively. The exhaust gases are mixed with the bypass stream through adjustable forward and rear VABI which regulate the bypass ratio to suit the overall cycle demands. For additional thrust on demand, more fuel is mixed with the turbine exhaust and re-combusted in the afterburner. The exhaust nozzle is equipped with variable convergent-divergent areas to provide controlled expansion of the exhaust gases to ambient pressure while maintaining proper back pressure to the engine [7]. Although other engines may vary slightly from this configuration, the concepts discussed here can be extended with minimal effort.

The MAPSS GUI, shown in Figure 4, provides a way for the designer to run the jet engine through various operating points, change engine constants including health parameters, plot any signal of interest, and save and compare the results. It was completely rewritten to automate the process of simulation and testing of new control laws. The GUI also automatically loads and saves initial conditions for the model and allows the user to setup several simulation runs at one time, allowing them to run sequentially and unattended over night. A built-in function allows the designer to create linear models for several operating points at once, run them in place of the full engine model, and compare the results with that of the corresponding nonlinear simulation.



MAPSS: Modular Aero-Propulsion System Simulation

MAPSS: Modular Aero-Propulsion System Simulation
NASA Glenn Research Center, Cleveland, OH

Simulation Running....

Profile Description: Cruise conditions with a nominal engine

Simulation Duration: 50 sec.
Engine Sample Time: 0.0004 sec. (fixed)
Controller Sample Time: 0.02 sec. (fixed)
Output Sample Time: 0.05 sec. (or Decimation Value)

Simulation Block: ------ CAD Output ------
Variable Name: Pick Component First

User Outputs:	Index
xnl_dot	159
xnh_dot	160
tmpc_dot	161
fn.	162
eprs	163
lepr	164
etr	165
sm2.	166
pcn2r	167
cepr	168

Results List:

	Operating Conditions				
Run #	PC	Mach #	Altitude	Duration	Constants Changed
Run1	30–33	0.8–0.8	36089–36089	100.0	0
Run2	33–36	0.8–0.8	36089–36089	100.0	0
Run3	36–39	0.8–0.8	36089–36089	100.0	0
Run4	39–42	0.8–0.8	36089–36089	100.0	0
Run5	42–45	0.8–0.8	36089–36089	100.0	0
Run6	45–48	0.8–0.8	36089–36089	100.0	0

Figure 4: MAPSS Graphic User Interface

2.3 Jet Engine Modeling

Obtaining an accurate model is often an essential step prior to the derivation of effective control laws. The jet engine model can be represented at various stages of complexity. In this section, the development of the jet engine model is discussed along with the significance of its different forms; a large scale model, a component level model, and a set of linear models.

The modeling equations of the turbofan have grown increasingly complex as the engine itself has evolved over time. At first, modeling was performed by incorporating large scale aerodynamic simulations to include complex component performance representations and precise thermodynamic calculations [7]. As the engine improved, the full range of operation, for which the model had to be accurate, has expanded multi-dimensionally. With the operating point represented by power lever angle (*pla*), altitude (*alt*), and Mach number (*xm*), the full envelope of operation now typically ranges from ground idle to maximum afterburner, from sea level to 60,000 feet altitude, and from static conditions to around mach 1.6.

2.3.1 The Component Level Model

Many control techniques mandated the use of a real time model to be present on an airborne digital controller, but the model had become too large to be useful in real time, even on the most sophisticated computers of the day. This has generated the following guidelines for a compact model [7]:

1. Solution accuracy over the full envelope of operating points.
2. All equations expressed as polynomials less than fourth order.
3. All equations run with a processor time step of 0.01 seconds without iteration between time steps.
4. A relatively low frequency model was allowed, as long as it permitted precise solutions to either time variant or steady state input values [8].

To meet these needs, GE developed a component level model (CLM), where each component of the engine such as fan, compressor, burner, turbine, nozzle, and bypass duct are modeled individually and then assembled together [7]. The fan inlet ducts were considered part of the airframe and consequently were not modeled as part of the CLM. A simplified top level diagram of the CLM used in MAPSS is shown in Figure 5.

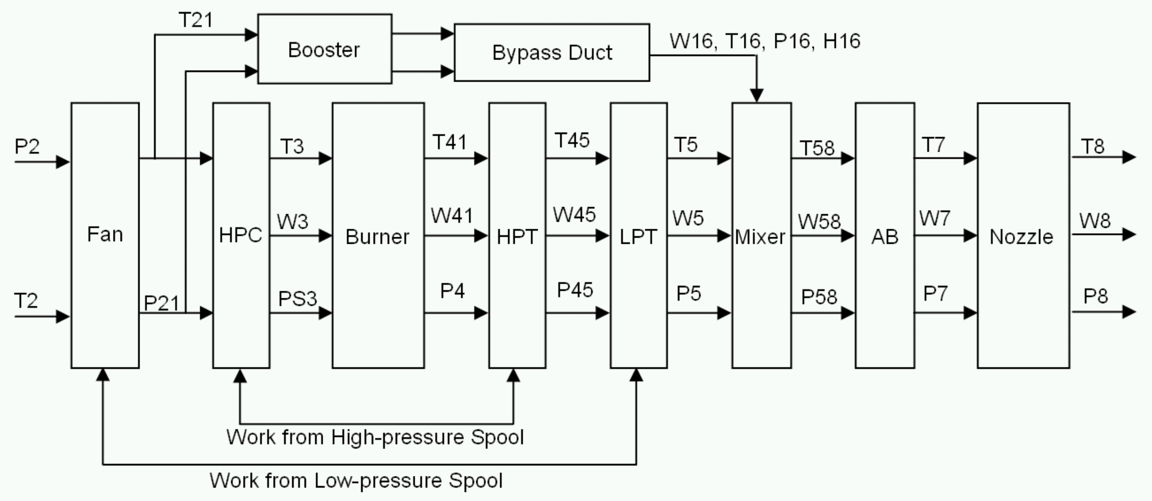


Figure 5: Engine Components

This approach was useful to represent the physics of the engine accurately in a relatively compact form [9]. That is, it is compact with respect to the full scale model yet still consists of hundreds coupled equations and look-up tables that ensure mass, momentum, and energy balances throughout while modeling gas properties effectively. Mathematical details can be found in books [2], [10], [11]. Each component consists of many inter-connecting blocks several levels deep. For example, the fan is represented by eight major components as well as five more stratification components that model the duct, core, and bypass streams.

The CLM is also capable of representing variation in component characteristics through the use of adjustable parameters. The key benefit of the CLM is that it is 140 times faster computationally than the most detailed aero-thermal cycle deck models, while maintaining accuracy within two percent [7], [9]. Recent advances in computing power have skyrocketed. However, the advances that have occurred in propulsion system technology, such as the ever-increasing complexity of variable-cycle engines, thrust vectoring nozzles, and the use of multiple nozzles from a single engine core, have sustained the need for the CLM.

Although the equations vary with engine type and application, as in [8], [9], [12]-[19], the CLM is generally defined by two nonlinear vector equations

$$\begin{aligned} \dot{x} &= f(x,u,t) \\ y &= g(x,u,t) \end{aligned} \tag{2.1}$$

where $x \in R^3$, $y \in R^{22}$, $u \in R^7$ in MAPSS. There are seven inputs, each controlled by a separate servo-mechanical actuator.

$$u = [wf36, a8, a16, a14, stp2, stp27, stp27d]^T \tag{2.2}$$

The outputs are defined by reliably sensed variables, such as the rotor speeds, pressures, and temperatures. Of the 22 sensor outputs, only twelve are of particular interest here.

$$y = [xn2, xn25, fn, sm2, pcn2r, t56, p16qp56, ps56, p27, t27, ps3, t3]^T \tag{2.3}$$

A vector of ten health parameters, denoted as *p*, defines the engine's relative age and state of degradation with efficiency and flow scalars of the turbo components. This vector is used to model the engine's time variance properties throughout its life span. The environmental conditions placed on the engine are defined by *alt* and *xm*, affecting the fan inlet pressure and temperature (*p2*) and (*t2*) and mass flow into the engine (*w2*). These parameters along with *pla* are used to model the engine's nonlinear properties throughout its full range of operation.

The states are defined by the two measured rotor speeds and the average metal temperature across the burner section (*tmpc*).

$$x = [xn2, xn25, tmpc]^T \tag{2.4}$$

The third state is a function of two unmeasured variables, *w3* and *t4*.

$$tmpc(s) = \frac{.5(t3 + t4)}{(498 / w3)s + 1} \tag{2.5}$$

Over the years, the states have been reduced from twenty or thirty to just three. This configuration was made possible through the aid of modern computing power and detailed studies of control modes for new and deteriorated engines. The objective was to select the slowest modes which indirectly control thrust while maintaining adequate margins and limits [8]. Time constants associated with heat transfer range in minutes, while time constants associated with the inertia of the two rotating shafts range in seconds. Other time constants associated with gas dynamics are less than one millisecond and can be neglected [4]. Although this configuration is not physically meaningful directly, it has the strongest relationship to the performance specifications. Furthermore, an intuitive input-output relationship exists. Fan speed *xn2* affects airflow and, consequently, thrust. Together *xn2* and *xn25* influence the bypass ratio, impacting specific fuel consumption, while *tmpc* sets the overall temperature ratio. Variations in *wf36*, *a8*, and *a16* have large effects on performance.

2.3.2 Linear Models

Although very accurate, the CLM has not been used to design engine control systems directly, due to the complexity of (2.1). Often linear state space models are used to design MIMO controllers. The linearization of (2.1) leads to a perturbation model describing the dynamic behavior of the engine around a single operating point, dictated by (*alt, xm, pla*).

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta u + B_1\Delta p \\ \Delta y &= C\Delta x + D\Delta u + D_1\Delta p \end{aligned} \tag{2.6}$$

In this equation, A, B, B_1, C, D, and D_1 are matrices and the perturbations are $\Delta x = x - x_0$, $\Delta u = u - u_0$, $\Delta p = p - p_0$, and $\Delta y = y - y_0$.

The process of linearization to determine each of the matrices is as follows. See reference [3] for more details. First, the CLM is run in closed loop at one operating point until it reaches steady state. The steady state values of x, u, p, and y are recorded and used as initial conditions in all subsequent open loop runs. At time t_0, the CLM is switched to open loop and run momentarily. The values of y, u, p, x, and $\dot{x}$ are recorded as variables y_0, u_0, p_0, x_0, and $\dot{x}_0$.

<u>*Linearization with Respect to States*</u>

The A and C matrices are determined by perturbing the states, one at a time. At time t_0, the CLM is again run open loop while the i^{th} state is perturbed by a step of Δx_i. The transient values of y, x, and $\dot{x}$ are recorded as variables y_1, x_1, and $\dot{x}_1$. For the sake of simplicity, determination of the A matrix is used as an example.

$$\begin{bmatrix}\Delta\dot{x}_1 & \cdots & \Delta\dot{x}_n\end{bmatrix} = A\begin{bmatrix}\Delta x_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta x_n\end{bmatrix} = \begin{bmatrix}A_{11}\Delta x_1 & \cdots & A_{1n}\Delta x_n \\ \vdots & & \vdots \\ A_{n1}\Delta x_1 & \cdots & A_{nn}\Delta x_n\end{bmatrix} \qquad (2.7)$$

The change in the $\dot{x}$ vector due to a change in x_i at t_0 is denoted as $\Delta\dot{x}_i$. From (2.7), it is clear that the i^{th} column of A can be calculated as $\Delta\dot{x}_i / \Delta x_i$. As a result, the i^{th} column of the *A* and *C* matrices are calculated as

$$A(:,i) = \frac{\dot{x}_1(t_1) - \dot{x}_0(t_1)}{\Delta x_i} \qquad C(:,i) = \frac{y_1(t_1) - y_0(t_1)}{\Delta x_i}, \qquad (2.8)$$

where t_1 denotes one time step after t_0. This is then repeated for each of *n* states.

<u>*Linearization with Respect to Control Inputs*</u>

The *B* and *D* matrices are determined by perturbing the inputs, one at a time. At time t_0, the CLM is run open loop while the i^{th} input is perturbed by a step of Δu_i. The transient values of *y* and $\dot{x}$ are recorded as variables y_2 and $\dot{x}_2$. The i^{th} column of the *B* and *D* matrices are calculated as

$$B(:,i) = \frac{\dot{x}_2(t_1) - \dot{x}_0(t_1)}{\Delta u_i} \qquad D(:,i) = \frac{y_2(t_1) - y_0(t_1)}{\Delta u_i}. \qquad (2.9)$$

This is repeated for each input.

<u>*Linearization with Respect to Health Parameters*</u>

The B_1 and D_1 matrices are determined by perturbing the health parameters, one at a time. At time t_0, the CLM is run open loop while the i^{th} health parameter is perturbed by a step of Δp_i. The transient values of *y* and $\dot{x}$ are recorded as variables y_3 and $\dot{x}_3$. The i^{th} column of the B_1 and D_1 matrices are calculated as

$$B_1(:,i) = \frac{\dot{x}_3(t_1) - \dot{x}_0(t_1)}{\Delta p_i} \qquad D_1(:,i) = \frac{y_3(t_1) - y_0(t_1)}{\Delta p_i}. \tag{2.10}$$

This is repeated for each health parameter.

Constructing a Working Model

Once the matrices are determined, a working perturbation model is constructed.

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B(u - u_0) + B_1(p - p_0) \\ y &= C\Delta x + D(u - u_0) + D_1(p - p_0) + y_0 \end{aligned} \tag{2.11}$$

Due to the nonlinearity of the CLM, a linear model accurately represents the characteristics of the engine in the vicinity of only a single operating point. Therefore, the engine is characterized over its full range by a set of linear models, one representing the engine at each critical operating point. A separate controller is designed for each model, and then the gains of all controllers are scheduled by curve fitting or optimization methods to form a full flight controller.

2.3.3 Additional Components

Additional components external to the CLM are modeled separately. They are discussed because they are part of the plant, and include performance parameters, actuators, open loop schedules, and combustor dynamics. A functional diagram of the plant is pictured in Figure 6.

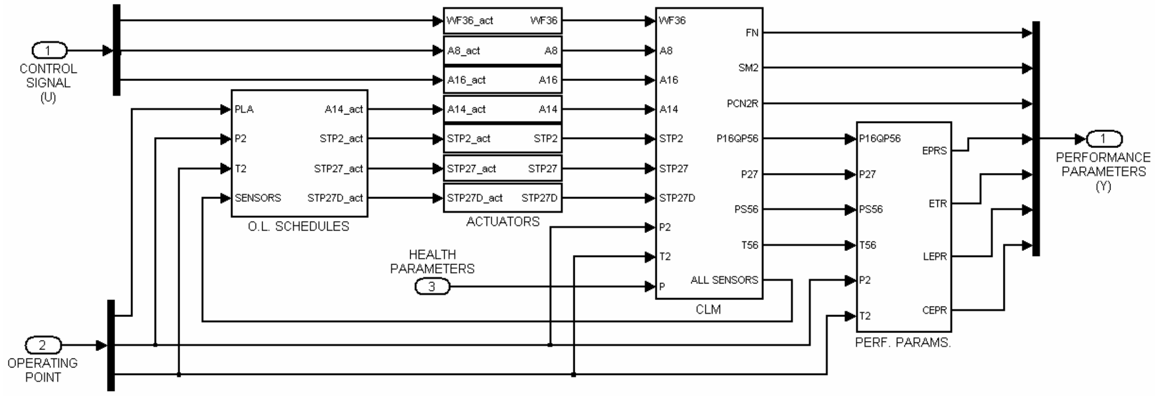


Figure 6: Functional Plant Diagram

The CLM sensor output vector *y* is combined to calculate vital pressure and temperature ratios. The engine pressure ratio (*eprs*) and engine temperature ratio (*etr*) are the ratios of static pressure and temperature across the engine. The core and liner engine pressure ratios (*cepr*) and (*lepr*) are the ratios of static pressure across the engine core and mixer, respectively. A model of these calculations is shown in Figure 7.

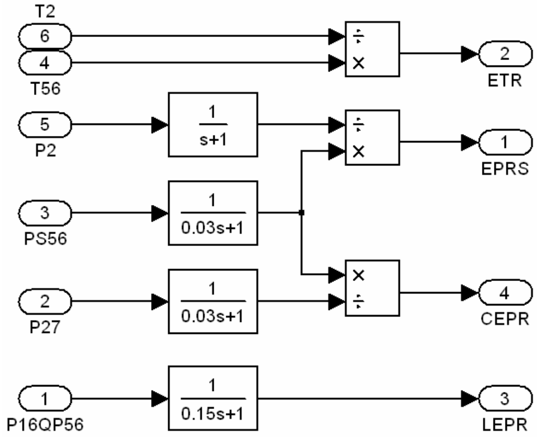


Figure 7: Performance Parameters

It is not uncommon for thrust (*fn*), the fan stall margin (*sm2*), the mixer pressure ratio (*p16qp56*), and the percent corrected fan speed (*pcn2r*) to be calculated here as well, since they cannot be measured directly by a sensor. However, in this particular engine, they are identified as CLM sensor outputs. For example, *p16qp56* is typically calculated as a ratio of the mixer inlet pressure (*p16*) and the LPT exit pressure at the mixer (*p56*) and *pcn2r* is calculated as a function of fan speed (*xn2*) corrected by the fan inlet temperature (*t2*). These performance parameters form the plant output.

$$Y = [fn, sm2, pcn2r, lepr, eprs, cepr, etr]^T \tag{2.12}$$

Each of the seven CLM inputs is controlled by a separate actuator consisting of a torque motor and servomechanism with saturation limits for position, velocity, and current. An actuator model is shown in Figure 8.

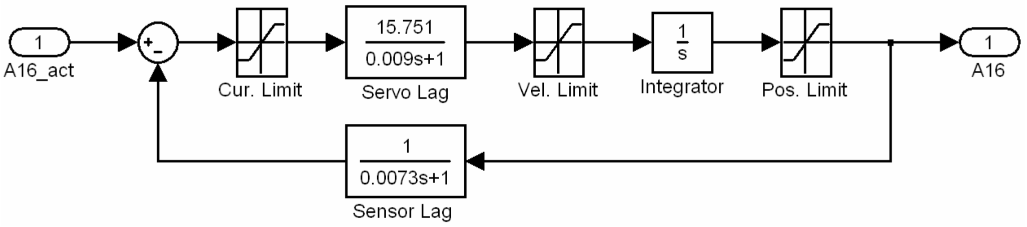


Figure 8: An Actuator Model

The first three actuators drive the fuel flow (*wf36*), variable nozzle exit area (*a8*), and rear bypass door variable area (*a16*), respectively. These actuator inputs form the plant input or control signal.

$$U = [wf36_{ACT}, a8_{ACT}, a16_{ACT}]^T \tag{2.13}$$

The remaining four actuators drive stator and guide vane angles using steady state schedules within the primary control loop, ensuring safe operating limits.

$$\begin{aligned} a14_{ACT} &= f_1(pcn2r) &&= f_1(xn2, t2) \\ stp2_{ACT} &= f_2(eprs) &&= f_2(ps56, p2) \\ stp27_{ACT} &= f_3(pnrm27) &&= f_3(xn25, t27) \\ stp27d_{ACT} &= f_4(pclm) &&= f_4(ps3, pla) \end{aligned} \tag{2.14}$$

The percent corrected core rotor speed (*pnrm27*) is a function of *xn25* and *t2*, and the limited power code (*pclm*) is the *pla* command after it has been rate limited by *ps3*. The stator and guide vane angles are usually kept on steady state schedules within the primary control loop. Furthermore, they are reflected in the linear model and help to ensure safe operating limits. To complete the full plant model, combustor delay and bypass dynamics are represented by a simple lag and time delay. The plant can then be represented as a nonlinear vector function in the form

$$\dot{Y} = F(Y, U, t) \tag{2.15}$$

where $Y \in R^7$, $U \in R^3$.

2.4 Control Objectives

A modern after-burning turbofan engine is a very complex power-plant with a wide range of operation. Furthermore, multiple objectives of control can present a real challenge to the control specialist [7]. The primary objective for a low bypass turbofan is to achieve the fastest possible thrust response during acceleration and deceleration. That is, the engine control system must provide the thrust demanded by the pilot through a change in the *pla* regardless of internal and external disturbances. At the same time, the control system must keep all engine variables within predefined limits [4], i.e. stall margins and maximum rotor speeds, pressures, and temperatures. External disturbances influence thrust and must be compensated. Environmental conditions such as ambient temperature, pressure, and humidity change significantly during flight [4]. The power take-off and bleed air extracted from the engine that provides power to external systems also creates external disturbances. Changing health parameters creates internal disturbances by changing various efficiencies and flows within the engine [4].

The aircraft manufacturer usually defines further control requirements, like the maximum allowable thrust overshoot (typically two percent) or the maximum allowable steady state thrust deviation (typically one percent) [4]. This is satisfied by designing a controller with the highest possible output bandwidth for a calculated thrust response before showing signs of overshoot. The bandwidth is limited by the open-loop dynamics of the engine and the level of complexity of the controller. One way zero steady-state error is achieved is by the use of integral gains in a multivariable PI controller [8].

Secondary objectives, outside the scope of this research, are to increase operating efficiency by minimizing variables such as specific fuel consumption (*sfc*) or to increase engine life and safety by adhering to stricter limits. Depending of the mission, two contradictory objectives can pose design trade-offs. An example of this is attempting to maximize one variable, i.e. thrust response, while minimizing other variables.

A functional diagram of the control configuration implemented in MAPSS is shown in Figure 9. The operating points, health parameters, and other constants are loaded from the GUI and the controller is run at a fixed sampling rate of 50 Hz. The plant is run at a much faster sampling rate, needed to iterate mass-energy balancing solutions. Limited by actuator time constants, this rate is 2500 Hz [6]. The inner loop is used to control integrator windup within the controller.

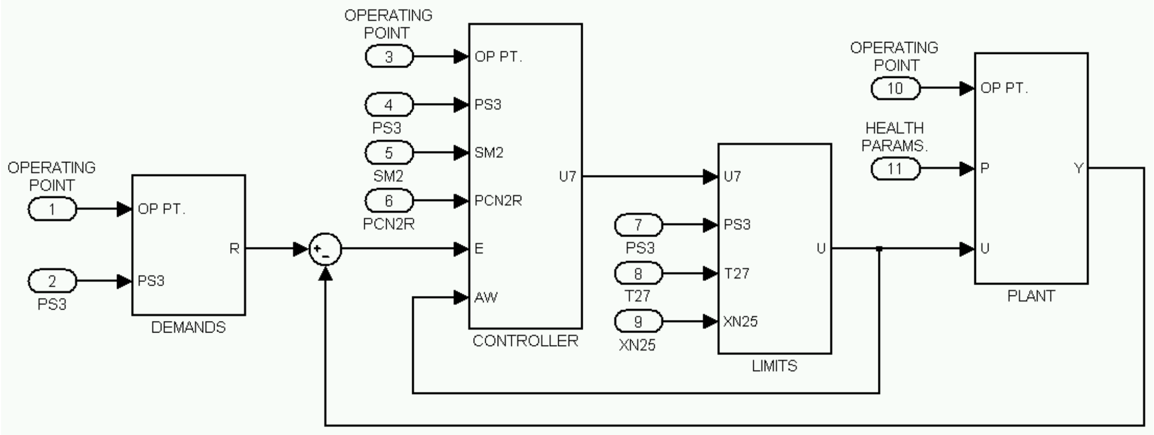


Figure 9: Control Configuration

2.5 Engine Constraints

Ideally, there would be no delay between the thrust commanded by *pla* movement and the net thrust delivered by the engine. However, this is not possible due to various operating limits [4]. Typical operational limits for a turbofan engine are fan and compressor stall margins, maximum fan and core rotor speeds, actuator limits, maximum turbine temperature, maximum combustor pressure, and maximum and minimum fuel-to-air ratios. Within the engine core, pressures and temperatures are often extreme and only *ps3*, *t27*, and *xn25* can be measured reliably [12]. Therefore, limits associated with the engine core are controlled by limiting the *wf36* input as functions of these measurable variables. These fuel flow limits impose acceleration and deceleration limits on the core rotor. Often critical fan variables can be measured reliably and, as a result, limits associated with the fan are controlled using closed-loop feedback.

The core rotor acceleration and deceleration is a function of fuel flow. Therefore, *wf36* is limited by an acceleration schedule to prevent the minimum compressor stall margin and the maximum turbine inlet temperature from exceeding their safe operating limits transiently. It also protects against primary combustor rich blowout. During maneuvers that require high acceleration, a stall of the flow around the compressor blades can cause permanent damage of the engine and must be avoided. The compressor surge margin is a measure of how close the compressor is to this type of instability. To prevent damage to the turbine blades, a thermal limit is placed on the temperature of the combustion chamber. Furthermore, flight authorities like the FAA pose certain demands on engine acceleration times, which can be extremely important in an emergency situation [4]. In MAPSS, the core rotor acceleration limit is implemented by placing a maximum bound on *wf36* as a function of *ps3*, *t27*, and *xn25*, preventing compressor surge, maximum turbine temperature, and combustor rich blowout.

$$wf36_{ACC} = f_5(ps3, t27, xn25) \tag{2.16}$$

Fuel flow is limited by a deceleration schedule to provide protection against primary combustor lean blowout [8]. It also provides constant deceleration times for similar engines [4]. In MAPSS, the core rotor deceleration limit is implemented by placing a minimum bound on *wf36* as a function of *ps3*.

$$wf36_{DEC} = 10\max(\frac{ps3}{.03s+1}, 50) \tag{2.17}$$

The following lowest/highest wins algorithm is then used to place these limits on the fuel flow commanded by the controller ($wf36_{ACT}$) before it is passed on to the actuation system [4].

$$wf36_{ACT} = \max(wf36_{DEC}, \min(wf36_{ACC}, wf36_{U7})) \tag{2.18}$$

Figure 10 depicts the fuel flow limits for the core rotor acceleration and deceleration. Actuator limits are also modeled to limit the outgoing control signal (*U*) because it is also fed back within the controller, Figure 9, to supply an integrator anti-windup scheme.

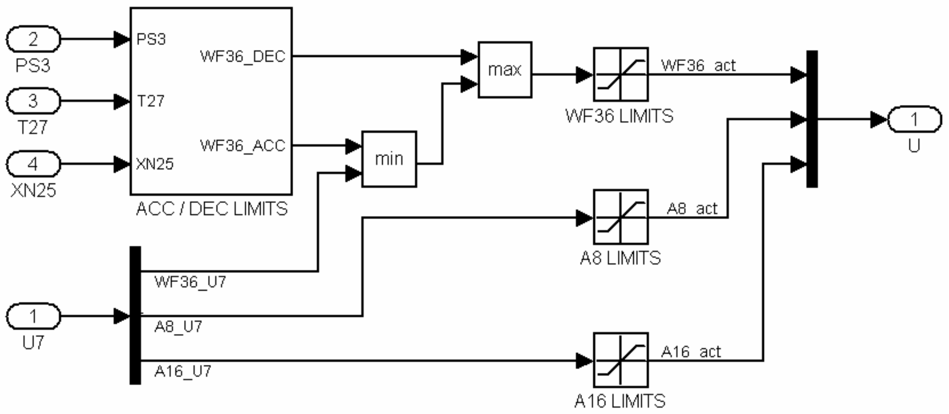


Figure 10: Control Signal Limits

The performance demands in Figure 9 are all calculated as functions of the current operating point, namely *pclm* which is the *pla* command after it has been rate limited by *ps3*. This rate limit helps to maintain engine core limits as well.

$$R = \begin{bmatrix} fn_{DEM} \\ sm2_{DEM} \\ pcn2r_{DEM} \\ lepr_{DEM} \\ eprs_{DEM} \\ cepr_{DEM} \\ etr_{DEM} \end{bmatrix} = \begin{bmatrix} f_6(pclm, alt, xm) \\ 10 \\ f_7(pclm, t2) \\ 1.05 \\ f_8(pclm) \\ f_9(pclm, t2) \\ f_{10}(pclm, t2) \end{bmatrix} \tag{2.19}$$

Limits associated with the fan are regulated within the controller itself. The minimum fan stall margin is another critical limit in turbofan engines. Though it cannot be directly measured, it can be correlated as a function of pressure ratio. If the total exit area decreases too quickly, backpressure builds up across the fan and the pressure ratio drops. The limit protection scheme will then increase the total exit area to maintain the pressure ratio, and thus the fan stall margin, above the minimum value [12]. In order to maintain structural integrity of the engine, a mechanical limit is placed on the maximum fan speed. The limit protection scheme will adjust the inputs to maintain *pcn2r* below its maximum value.

2.6 Multi-Mode Control

The error signal into the MAPSS controller is represented by

$$E = R - Y = \left[fne, sm2e, pcn2re, lepre, eprse, cepre, etre\right]^T \tag{2.20}$$

and the control signal out of the controller is represented by

$$U_7 = \left[wf36_{U7}, A8_{U7}, A16_{U7}\right]^T . \tag{2.21}$$

Since $E \in R^7$ and $U_7 \in R^3$, the controller consists of four separate regulators, each with an error signal $E_j \in R^3$, $j=1,2,3,4$. This allows for square invertible matrices in the regulators, ideal for decoupling to take place.

$$\begin{aligned} E_1 &= [fne, eprse, lepre]^T \\ E_2 &= [fne, etre, lepre]^T \\ E_3 &= [fne, sm2e, lepre]^T \\ E_4 &= [pcn2re, cepre, lepre]^T \end{aligned} \tag{2.22}$$

The first regulator controls *eprs* at low speeds, while the second regulator controls *etr* at high speeds. The third and fourth regulators actively control limits associated with the fan components, namely fan stall margin and fan over-speed when their limits are approached. To ensure zero steady state error, multivariable PI controllers in the form of

$$U_j = (K_{jP} + K_{jI}(sI)^{-1})E_j \tag{2.23}$$

are implemented where

$$K_{jP} = \begin{bmatrix} g_{j1} & g_{j2} & g_{j3} \\ g_{j4} & g_{j5} & g_{j6} \\ g_{j7} & g_{j8} & g_{j9} \end{bmatrix} \quad K_{jI} = \begin{bmatrix} g_{j10} & g_{j11} & g_{j12} \\ g_{j13} & g_{j14} & g_{j15} \\ g_{j16} & g_{j17} & g_{j18} \end{bmatrix}. \tag{2.24}$$

Across the four regulators, each of the 72 gains are scheduled with functions of *pcn2r*, *p2*, *t2* and six constants

$$g_{ji} = c_{ji1} + c_{ji2}\exp[c_{ji3} + c_{ji4}f_{14}(pcn2r) + c_{ji5}f_{15}(t2) + c_{ji6}f_{16}(p2)] \tag{2.25}$$

where $j \in R^4$, $i \in R^{18}, h \in R^6$, $c_{jih} \in R^{432}$. This produces 432 parameters that require proper adjustment. The regulators are implemented with a discrete time step *dt=.02 sec.* using backward Euler integration.

$$\begin{aligned} U_j(k) &= I_j(k) + K_{jP}E_j(k) \\ I_j(k) &= I_j(k-1) + K_{jI}\frac{E_j(k)+E_j(k-1)}{2}dt + U(k-1) - U_j(k-1) \end{aligned} \tag{2.26}$$

Here, the integral term *I_j(k)* consists of an accumulator term and an input term, followed by two terms that control integrator windup. The input term averages the last two error samples.

There are three control modes which govern the operation of the four regulators and the combination of their control signals. At *pla* values below 37.5, the low speed regulator is fully operational. Above this point, the high speed control mode

$$\alpha_1 = \begin{cases} 0, & pclm < 37.5 \\ \dfrac{pclm - 37.5}{5}, & otherwise \\ 1, & pclm > 42.5 \end{cases} \tag{2.27}$$

begins to take affect when *pla* reaches a value of 37.5, and becomes fully operational at values above 42.5. The stall margin control mode

$$\alpha_2 = \begin{cases} 0, & sm2 > 10 \\ \dfrac{10 - sm2}{5}, & otherwise \\ 1, & sm2 < 5 \end{cases} \tag{2.28}$$

begins to take affect when *sm2* falls below a value of 10, and becomes fully operational at values below 5. The fan over-speed control mode

$$\alpha_3 = \begin{cases} 0, & pcn2r < 100 \\ \dfrac{pcn2r - 100}{5}, & otherwise \\ 1, & pcn2r > 105 \end{cases} \tag{2.29}$$

begins to take affect when *pcn2r* reaches a value of 100, and becomes fully operational at values above 105. The four controllers are then blended in linear combination, where α_2 supersedes α_1, and α_3 supersedes α_2.

$$\begin{aligned} U_5 &= (1-\alpha_1)U_1 + \alpha_1 U_2 \\ U_6 &= (1-\alpha_2)U_5 + \alpha_2 U_3 \\ U_7 &= (1-\alpha_3)U_6 + \alpha_3 U_4 \end{aligned} \tag{2.30}$$

The control signal then becomes

$$U_7 = [[(1-\alpha_1)U_1 + \alpha_1 U_2](1-\alpha_2) + \alpha_2 U_3](1-\alpha_3) + \alpha_3 U_4 . \tag{2.31}$$

After combining terms, it is represented as a linear combination of the control signals from each of the four regulators

$$U_7 = \beta_1 U_1 + \beta_2 U_2 + \beta_3 U_3 + \beta_4 U_4 \tag{2.32}$$

where

$$\begin{aligned} \beta_1 &= 1\ (1-\beta_2-\beta_3-\beta_4) \\ \beta_2 &= \alpha_1(1-\beta_3-\beta_4) \\ \beta_3 &= \alpha_2(1-\beta_4) \\ \beta_4 &= \alpha_3 \end{aligned} \quad . \tag{2.33}$$

Interestingly $0 < \beta_{i\in 4} < 1$ and $\sum \beta_i = 1$. A block diagram of the entire controller is shown below in Figure 11.

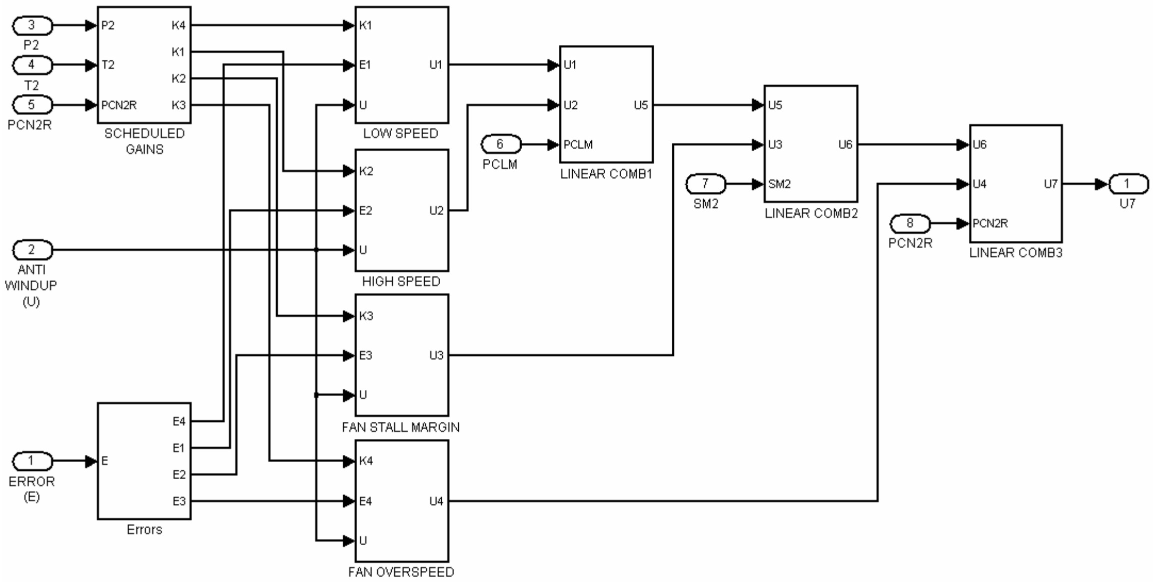


Figure 11: Multi-Mode Controller

CHAPTER III

LITERATURE REVIEW

Various methods have been proposed to control turbofan engines, but only one has been successfully transferred to technology. In an attempt to understand the primary reasons why many of the control methods do not work in practice, this chapter briefly examines the basic concepts behind each method. The chapter concludes with a brief introduction to potential control methods that contain elements needed to produce a practical solution.

Although the CLM is compact and capable of generating accurate results, it is highly nonlinear and generates widely varying results over the full flight envelope. However, most multivariable control techniques are determined from a set of linear state space models. As a result, two methodologies have been followed to design controllers for turbofan engines and other complex systems; the current approach that is outlined in Section 3.1, and other modern methods that are outlined in Section 3.2.

3.1 Proportional Integral Derivative

The PID structure is presently used in 95 percent of all industrial control applications. Most jet engines in production employ some sort of multivariable PID controller because the design approach is error-based, allowing the controller parameters to be adjusted by watching an available signal. For low power levels close to flight idle operation, single-input single-output (SISO) compensators are used [8]. Three-input three-output multivariable PI compensators are designed for the higher power levels between 50 to 100 percent of maximum dry power where they are most effective.

3.1.1 Present-Day Turbofan Control

The current design method is summarized in three basic steps. The first step is to reduce the CLM to a linear state space model at a single operating point, and then to derive a linear control law for it. The linear control design is obtained using classical SISO techniques [14], [15] or a nonlinear constrained optimization algorithm to minimize specific fuel consumption as a function of model inputs. The engine limits and stall margins are then considered as constraints, assuring a safe optimum [8]. The second step is to repeat the modeling and control design process at significantly varying operating points over the full flight envelope. The third and final step is to schedule each of the control laws' gains through a suitable curve-fitting algorithm or table-lookup scheme which produces an operational controller for the entire flight envelope [13]. The schedules at critical operating points must be treated as independent parameters in the curve-fitting routine to ensure suitable performance at such points [14]. This engineering approach has been successfully applied to currently operating aircraft and engines with control laws based on classical SISO techniques [14].

In [20], a model matching technique was proposed to determine gains of multivariable PID controllers. It is a frequency-domain technique that uses closed-loop Nyquist and Bode plots. The incorporation of singular value ideas ensures good fundamental feedback properties. In general, parameters of a dynamic matrix compensator K are estimated by minimizing the mean square error between the desired closed-loop response Q and the actual closed-loop response. Also referred to as the KQ method, it was applied in [8] to control a turbofan engine for full flight operation with power level variations. It consists of the following steps:

1. Selection of the MIMO compensator form (P, PI, PID, etc.)
2. Specification of desired closed-loop transfer functions matrix.
3. Computation of compensator gains at each operating point
4. Curve-fit each set of gains to produce a full-flight controller
5. Conduct a closed-loop performance check

As previously mentioned, turbofan engine control specifically requires fast thrust response without overshoot, and zero steady-state error. A multivariable PI controller is used because it is the simplest form to meet such requirements [20], [8].

Although the controller can be tuned, the problem becomes the sheer number of parameters compounded by scheduling. A three-input three-output controller with 18 gains per operating point poses difficulties in tuning because it is not apparent how to adjust the gain schedules when the overall performance is unsatisfactory on a real engine.

Furthermore, the KQ technique does not guarantee stability or take into account actuator rate limits [15]. Despite these shortcomings, it is still the control method of choice on currently operating aircraft.

3.1.2 Nonlinear PID

Although PID is error-based and has three tuning parameters, it is limited in performance by its linear gains. A nonlinear PID technique was recently introduced in an attempt to increase performance. A brief overview is first given, followed by a discussion of its strengths and weaknesses. This section concludes with a review of other nonlinear forms.

Algorithm

Consider the PID control law where e is the error signal.

$$u = k_p e + k_i \int e\,dt + k_d \dot{e} \tag{3.1}$$

Since the proportional term cannot affect small errors enough to drive them to zero without saturating for large errors, a nonlinear function was proposed to replace k_p, making the gain small for large errors to minimize overshoot and saturation and large for small errors to minimize the steady state error [21], [22]. This reduces the need for integral and derivative terms which help to correct the impact of the linear proportional term in the first place. Note that motion profiles are effective because they keep the error in a small range, but they are only effective on the initial transient.

The integral term is most effective in the steady state region when the error is small. Here it ensures that the otherwise finite error is driven to zero. Other places in the response, however, the integral is prone to saturation and windup. There is also a 90° phase lag associated with the integrator that can push higher-order plants towards instability. As a result, a nonlinear function was proposed to replace k_i, making the gain zero when the error is not small [21], [22]. Engine and actuator limit protection schemes often call for integrator windup protection (IWP). The nonlinear integral term may prove beneficial, since it operates in a similar fashion to the engine and actuator limits and is built right into the controller.

The derivative term is most effective in the transient region where the error is changing rapidly, and least effective in the steady state region where it basically processes noise and degrades the control. As a result, a nonlinear function was proposed to replace k_d, making the gain zero when the error is small [21], [22]. Sometimes the derivative term is not useful because the manner in which the derivative itself is extracted can amplify noise to unacceptable levels. In general, the derivative should be obtained in a noise-free fashion. Approximations are typically made using filters. However, the drawback is that additional tuning parameters are associated with each filter.

Strengths and Weaknesses

Nonlinear PID offers the robustness of modern control methods without the drawbacks of having to create linear models. It also addresses other real-world complications such as integrator phase lag and windup, the noise amplification of derivative terms, and the inaccuracy of derivative approximates. It has been demonstrated in several commercial applications and has proven to be a significant advancement in feedback control technology [23]-[26]. Despite the obvious advantages, the tradeoff for better performance is that the three PID parameters are replaced with nine. A three-input three-output jet engine model would require 54 gains for each scheduled operating point.

Other Nonlinear Forms

The tracking differentiator was proposed as a nonlinear alternative to the second order approximate derivative. It is a filter that is not dependent on the frequency composition of the system [21] and has only one design parameter *M* with direct physical significance to the maximum acceleration of the system [22].

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -M \; sign\left(x_1 - u + \dfrac{x_2 \, | \, x_2 \, |}{2M} \right) \end{cases} \tag{3.2}$$

By using time optimal control to control a double integral plant, the states x_1 and x_2 track the input signal *u* and its derivative, respectively.

Recently, a discrete time optimal control (DTOC) law was developed [27], [28]. Truly integrator-less, it is given as a function of the current time step

$$\begin{aligned} & d = Mh \\ & y(k) = r(k) - x_1(k) + h(\dot{r}(k) - x_2(k)) \\ & a_0(k) = \sqrt{d^2 + 8M\left|y(k)\right|} \\ & a(k) = \begin{cases} \dot{r}(k) - x_2(k) + \dfrac{a_0(k) - d}{2} sign(y(k)), & \left|y(k)\right| > dh \\ \dot{r}(k) - x_2(k) + y(k)/h, & \left|y(k)\right| \le dh \end{cases} \\ & u(k) = -\begin{cases} M \cdot sign(a(k)), & \left|a(k)\right| > d \\ M \cdot a(k)/d, & \left|a(k)\right| \le d \end{cases} \end{aligned} \tag{3.3}$$

where *r*(*k*) is the reference input, *u*(*k*) is the controller output, *h* is the discrete step size, and *M* is a tuning parameter. It was designed to control a discrete double integral plant

$$\begin{aligned} &x(k+1) = \Phi x(k) + \Gamma u(k) \\ &\Phi = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 \\ h \end{bmatrix} \end{aligned} \tag{3.4}$$

where $\left| u(k) \right| \leq M$ and $x(k) = [x_1, x_2]^T$. The robustness of DTOC has allowed it to be used as a nonlinear PD controller for second order plants in general [28]. It can also be used as a discrete tracking differentiator. By combining (3.3) with (3.4), the states $x_1(k)$ and $x_2(k)$ will track the input signal *r*(*k*) and its derivative, respectively.

3.1.3 Structural Drawbacks

Since its inception over eighty years ago [29], the PID design framework has remained the most commonly used SISO control technique in industry [30], primarily because its simplicity makes it easy to use. However, its simplicity also limits the overall performance [31]. Although various tuning techniques and external filters have been applied to the PID structure [32], the additional complexity leading to higher performance ultimately comes at the expense of usability. To develop a high performance solution that is still easy to use, the control law, structure, and existing tuning methods are addressed as a whole before a comprehensive solution is proposed.

Control Law Limitations

The PID controller itself is limited in performance, particularly when dealing with dominant dynamics higher than second order [32]. Processes with three or more equal lags and resonant modes fall into this category. Even dominant delays are often modeled using linear approximations which increase the overall order of the system [33]. To accommodate higher order systems, the PID control law is generalized.

Design Trade-off

A PID controller can be designed by simply adjusting a few parameters to produce a desired closed loop response. However, the single DOF structure creates a compromise in performance between disturbance rejection and command following because all design requirements must be satisfied through one mechanism [31]. Cancellation between the plant and controller produces uncontrollable and unobservable modes which lead to poor disturbance rejection at the cancelled frequencies [32]. Consequently, this approach often requires retuning.

To fix this problem, Ziegler-Nichols [34] and other tuning methods [32] were designed primarily to provide good disturbance rejection. A prefilter, lead-lag compensator, or set point weighting is then applied to form a more complex 2-DOF control structure and generate a desired closed loop response. To eliminate the need for independent prefilter design and tuning, a prefilter is proposed as a function of the generalized controller, making the design process transparent to the end user.

Modeling and Tuning Bottlenecks

Controller parameters are typically based on model information that was acquired through some sort of preliminary system identification technique. When linear time-invariant models are used to approximate physical systems that are nonlinear and time-varying in practice, the controller may require retuning as the system changes.

Furthermore, the design process is often reduced by assuming a simpler model with fewer parameters. When a control structure is robust, by definition it is able to account for these model inaccuracies and performance is affected only slightly. But when a control structure is not robust, then performance is largely dependent on the accuracy of the model. Therefore, current tuning methods are limited by the structure to which they are applied and require a preliminary test for tuning the controller.

3.2 Modern Control Techniques

The more recent approach to jet engine control design has been characterized by the use of modern design techniques. The idea is to design a robust controller that can function over a wider range of operation than traditional approaches. This would allow for the reduction of gain scheduling complexity or attempt to eliminate it altogether [32].

3.2.1 Turbofan Control Research

A great deal of research has been conducted towards the application of modern multivariable control techniques on aircraft engines. The majority of this research has been to control the engine at a single operating point. Among these methods are a multivariable integrator windup protection scheme [16], a tracking filter and control mode selection for model based control [5], an H_∞ method and linear quadratic Gaussian with loop transfer recovery method [13], and a performance seeking control method [9].

Various schemes have been developed to reduce gain scheduling [14] and were even combined with integrator windup protection and H_∞ as well [15].

There have been a limited number of control techniques for full flight operation [8], [14], but no logical way of tuning the controller for satisfactory performance when applied to an actual engine has yet been discovered. At any given operating point, models become inaccurate from one engine to another. This accuracy increases with model complexity, and subsequently design and tuning complexity.

Although the CLM is very accurate, it is not directly used to design a controller. This is because all available design methods are based on linear time-invariant (LTI) models. That is, even if a designer had perfect knowledge of a nonlinear time-varying (NTV) system, there is no way to use these equations directly to design a controller. Instead, linear models are created to design a controller and the design process becomes more complex to deal with this linearization, making real world tuning less attainable. As a result, very few of these or similar aircraft design studies have led to implementation on an operational vehicle [14].

3.2.2 Model Limitations

Advances in observer design throughout the years have brought about an overall increase in control system performance. Observers extract real-time information of a plant's internal state from its input-output data. The observer usually presumes precise model information of the plant [35]-[42], since performance is largely based on its mathematical accuracy. Closed loop controllers require both real-time information from an observer and predetermined model information. Such presumptions, however, often make the method impractical in engineering applications, since the challenge for industry remains in constructing these models as part of the design process. Another level of complexity is added when gain scheduling and adaptive techniques are used to deal with nonlinearity and time variance, respectively.

Derivative approximations can be considered simple observers in the way they filter plant output data to supply useful feedback [43]. For example, the $\alpha - \beta - \gamma$ filter discretely estimates position, velocity, and acceleration in this way [44], [45]. Open loop observers use model information and plant input data to estimate unmeasured states. The problem with open loop observers, however, is that a precise mathematical model must be known in order to achieve satisfactory performance. Closed loop or Luenberger observers combine model information with plant input and output data to improve state estimation in the presence of initial conditions [36], [38], [39]. Again, the assumption is that the system is represented by a known LTI model.

In general, the performance of model-based controllers is largely dependent on the accuracy of the model. When linear models are used to characterize nonlinear systems, the model information becomes inaccurate as the plant transitions away from the design operating point. When time invariant models are used to approximate time varying systems, the constant model information becomes inaccurate over time. As a result, gain scheduling and adaptive techniques are developed to deal with nonlinearity and time variance, respectively. However, the complexity added to the design process leads to an impractical solution for industry because of the time and level of expertise involved in designing, tuning, and maintaining each control system.

3.3 Disturbance Rejection Techniques

To account for uncertainties in the real world, various techniques have been proposed [46]-[52] that solve the problem of model accuracy in reverse. This is done by modeling a plant *P* with an equivalent input disturbance *d* that includes unmodeled dynamics. An observer is then designed to estimate the disturbance in real time and provide feedback to cancel it. As a result, the augmented system reduces to the nominal model P_n at low frequencies, making it accurate. This referred to as disturbance rejection and is depicted in Figure 12.

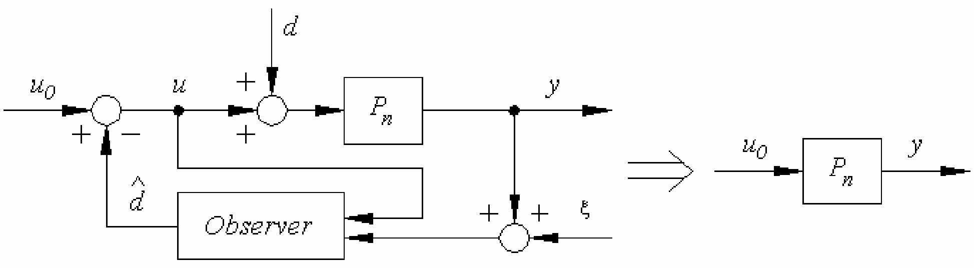


Figure 12: Disturbance Rejection

The most common of these techniques is the disturbance observer (DOB) framework [50], [53], [54]. It uses simple binomial Q-filters, allowing the observer to be tuned by a single bandwidth parameter. A model deliberately different from *P* is also suggested to facilitate design [54], but no guidelines are given other than it should be as simple as possible, cautioning stability and performance may be in danger. Another obstacle is that a separate observer must be designed to provide state feedback to the controller. In existing research, derivative approximates are used in this way but their affect on performance and stability has yet to be analyzed.

Another technique, referred to as the unknown input observer (UIO), estimates the states of both the plant and the disturbance by augmenting a linear nominal model with a linear disturbance model [40]-[42], [46]-[49], [55]. Unlike the DOB, a controller and observer can be designed independently, like a Luenberger observer. However, it still relies on a good mathematical model and a design procedure to determine observer gains.

Multivariable Cases

Multiple disturbance observers were used to control a multivariable robot by treating it as a set of decoupled SISO systems, each with disturbances that included the coupled dynamics [53], [54], [56], [57]. This technique proved to be very affective and is also used in the derivation of this research.

3.4 Active Disturbance Rejection Control

At the heart of ADRC is the ESO. The concept originated from a desire to provide practical solutions to industry, specifically to provide a control framework that is capable of high performance but is also easy to implement. Many control methods don't get transferred to technology because of the tradeoff between performance and complexity. Although disturbance techniques are a step in the right direction, there are two elements that set ADRC apart from the rest. The first is that the plant is represented in a unique canonical form and the second is that the controller and observer, including state estimates used by the controller, are a function of two adjustable bandwidths.

The Extended State Observer

The ESO is quite different from the UIO in that it does not rely on a rigorous mathematical model. Originally proposed in the form of a nonlinear UIO [51] and later simplified to a linear version with one tuning parameter [52], the ESO combines the state and disturbance estimation power of a UIO with the tuning simplicity of a DOB.

One finds a decisive shift in the underlying design concept as well. The traditional observer is based on a linear time-invariant model that often describes a nonlinear time-varying process. Although the DOB and UIO reject input disturbances for such nominal plants, they leave the question of dynamic uncertainty mostly unanswered in direct form. The ESO, on the other hand, addresses both issues in one simple framework by formulating the simplest possible design model P_d for a large class of uncertain systems. P_d is selected to simplify controller and observer design, forcing P to behave like it at low frequencies rather than P_n. As a result, the effects of most plant dynamics and external disturbances are concentrated into a single unknown quantity, representing any difference between P and P_d. The ESO estimates this quantity along with derivatives of the output, giving way to the straightforward design of a high performance controller. Its practical usefulness is seen in a number of benchmark applications already implemented throughout industry [24], [58]-[61].

A Second Order Case

Consider a second order plant of the following form

$$\ddot{y} = f(y, \dot{y}, w, t) + bu \tag{3.5}$$

where u and y are the input and output, respectively, and $f(y,\dot{y},w,t)$ is the combined external disturbance w and system dynamics. It is represented with state equations

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = \dot{f}(y,\dot{y},w,t) \\ y = x_1 \end{cases} . \tag{3.6}$$

where $[x_1, x_2, x_3] = [y, \dot{y}, f]$. An ESO is then constructed to estimate the states.

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + l_1(y - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu + l_2(y - \hat{x}_1) \\ \dot{\hat{x}}_3 = l_3(y - \hat{x}_1) \end{cases} \tag{3.7}$$

A control law is applied to (3.5) to dynamically cancel $f(y,\dot{y},w,t)$

$$u = (u_0 - \hat{x}_3)/b \tag{3.8}$$

reducing the plant to $\ddot{y} \approx u_0$ whereby a second control law is implemented.

$$u_0 = k_p(r - \hat{x}_1) - k_d\hat{x}_2 \tag{3.9}$$

By substituting (3.8) into (3.7), integrating $\dot{\hat{x}}_3$and combining it and (3.8) with (3.9), ADRC represented by (3.7) through (3.9) simplifies to an equivalent PID form with one less algebraic loop.

$$\begin{aligned} & bu = k_p(r - \hat{x}_1) - k_d\hat{x}_2 - \int l_3(y - \hat{x}_1) \\ & \begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + l_1(y - \hat{x}_1) \\ \dot{\hat{x}}_2 = u_0 + l_2(y - \hat{x}_1) \end{cases} \end{aligned} \tag{3.10}$$

The estimated output and its derivative are represented by a simple Luenberger observer of a double integrator. This motivates the research to question the connection between ADRC and PID and also to question how ADRC concepts can be used to improve PID, since PID is currently used in multivariable control, i.e. aircraft engines.

A Multivariable Case

In [62], a multivariable flight controller is designed by incorporating an ESO with nonlinear feedback. The aircraft equations are rewritten in the form

$$\begin{aligned} \dot{X}_1 &= F_1(X_1)X_2 \\ \dot{X}_2 &= F_2(X_1, X_2, X_3, X_4) + B(X_1, X_3, X_4)U \\ \dot{X}_3 &= F_3(X_1, X_4) \\ \dot{X}_4 &= F_4(X_1, X_2, X_3, X_4, U) \end{aligned} \tag{3.11}$$

where $X_1 = \begin{bmatrix} \vartheta \\ \psi \\ \gamma \end{bmatrix}$, $X_2 = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$, $X_3 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, $X_4 = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$,and $U = \begin{bmatrix} d_a \\ d_e \\ d_r \end{bmatrix}$. It is assumed that X_1, X_2, and a state V are the only directly measurable states. When $H(t)$ is defined as

$$H(t) \equiv F_2(X_1, X_2, X_3, X_4) + (B(X_1, X_3, X_4) - B_0(V))U\ , \tag{3.12}$$

and $B_0(V)$ is an approximation of $B(X_1,X_3,X_4)$, the system reduces to

$$\begin{aligned} \dot{X}_1 &= F_1(X_1)X_2 \\ \dot{X}_2 &= H(t) + B_0(V)U \end{aligned} . \tag{3.13}$$

At this point, a multivariable ESO is designed by arranging three SISO second order extended state observers in parallel such that Z_2 estimates $H(t)$ in real time

$$\begin{aligned} \dot{Z}_1 &= Z_2 + \beta_1(E_1) + B_0(V)U \\ \dot{Z}_2 &= \beta_2(E_1) \end{aligned} \tag{3.14}$$

where $E_1 = X_2 - Z_1 = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$, $\beta_1(E_1) = \begin{bmatrix} L_{11} \cdot e_1 \\ L_{11} \cdot e_2 \\ L_{11} \cdot e_3 \end{bmatrix}$, $\beta_2(E_1) = \begin{bmatrix} L_{21} \cdot f(e_1) \\ L_{22} \cdot f(e_2) \\ L_{23} \cdot f(e_3) \end{bmatrix}$. The control law,

$$U = B_0^{-1}(V)(U_1 - Z_2), \tag{3.15}$$

is applied, further reducing the system to

$$\begin{aligned} \dot{X}_1 &= F_1(X_1)X_2 \\ \dot{X}_2 &= U_1 \end{aligned} . \tag{3.16}$$

A non-smooth feedback control law

$$U_1 = \begin{bmatrix} k_{21}|e_{21}|^{\alpha 2} sign(e_{21}) \\ k_{22}|e_{22}|^{\alpha 2} sign(e_{22}) \\ k_{23}|e_{23}|^{\alpha 2} sign(e_{23}) \end{bmatrix}, \; X_2^* - X_2 = \begin{bmatrix} e_{21} \\ e_{22} \\ e_{23} \end{bmatrix} \tag{3.17}$$

incorporates a back-stepping function

$$X_2^* = F_1^{-1}(X_1)U_0 \tag{3.18}$$

where a second non-smooth feedback control law is then applied.

$$U_0 = \begin{bmatrix} k_{11}|e_{11}|^{\alpha 1} sign(e_{11}) \\ k_{12}|e_{12}|^{\alpha 1} sign(e_{12}) \\ k_{13}|e_{13}|^{\alpha 1} sign(e_{13}) \end{bmatrix}, \; X_1^* - X_1 = \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \end{bmatrix} \tag{3.19}$$

It is quite apparent that this solution is problem specific. Although a multivariable ESO is constructed to estimate the states of this system, a general method for uncertain MIMO systems remains to be proposed until now.

3.5 Motivation

There is often no logical way of tuning existing control structures to improve undesirable performance when applied to a specific jet engine. Model based methods rely on linear models that do not characterize NTV systems well. Gain scheduling creates too many tuning parameters, even with error based control methods. The nominal controller in MAPSS utilizes 432 individual parameters.

In ADRC, a plant is represented in a canonical form with a generalized disturbance that is composed of both internal uncertainties and external disturbances. Perhaps robustness and disturbance rejection are both achieved by overcoming this disturbance. A robust control structure is one that is less sensitive to dynamic uncertainty, such as engine-to-engine variation. A control structure with good disturbance rejection has the ability to account for external disturbances, such as slow engine degradation that is modeled as a constant disturbance.

The objective of this research is to develop a multivariable control framework with less tuning parameters than current methods, while maintaining or even improving performance.

CHAPTER IV

A NEW DISTURBANCE REJECTION FRAMEWORK

In this chapter, a number of control design tools are proposed, forming a framework for controlling NTV and uncertain systems. A discrete ESO, a reformulation of the PID structure, and a generalization of the ADRC structure are proposed. This framework is extended to systems of arbitrary order as well as to the general MIMO case.

4.1 The Extended State Observer

Enhancements are made to the ESO in this section, both in formulation and in implementation. First, the ESO is discretized as a current discrete estimator to maintain a bounded output at lower sampling rates, a major limiting factor in controls. Typical discretization methods, such as the predictive discrete estimator [40], generate at least one sample of delay, whereas the current discrete estimator removes this delay by adding a current time step update to the estimated state [40]. Next, both zero order hold (ZOH) and first order hold (FOH) versions of all discrete matrices are determined symbolically to retain single parameter tuning. In the past, only Euler integration was used. Finally, the ESO is reformulated to incorporate a disturbance model of arbitrary order, thus allowing the amount of disturbance rejection to be specified for different types of disturbances. In the past, disturbances were restricted to first order.

A current discrete ESO is formulated and analyzed in Section 4.1.1 using symbolic Euler and ZOH methods [63]. It is reformulated with multiple extended states for a system of arbitrary order in Section 4.1.2 using symbolic ZOH and FOH methods. A realistic simulation test of a motion control plant is also provided in Section 4.1.3 [63].

4.1.1 Discrete Implementation

For the sake of simplicity, consider a general second order plant where *u* and *y* are the input and output, respectively, and *b* is a constant.

$$\ddot{y} = g(y, \dot{y}, t) + w + bu \tag{4.1}$$

Combining the internal dynamics $g(y, \dot{y}, t)$ with an external disturbance w to form a generalized disturbance $f(y, \dot{y}, w, t)$, the system is rewritten as

$$\ddot{y} = f(y, \dot{y}, w, t) + bu \tag{4.2}$$

Continuous Observer Design

An augmented state space model is constructed

$$\begin{aligned} \dot{x} &= Ax + Bu + E\dot{f} \\ y &= Cx + Du \end{aligned} \tag{4.3}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

where $x = [y, \dot{y}, f]^T$ includes the disturbance to be estimated. Next, an observer is created from the state space model.

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} + Du \end{aligned} \tag{4.4}$$

Note that $\dot{f}$ is ignored in (4.4) since it is unknown and is estimated by the correction term. The observer is rewritten to output the state

$$\begin{aligned} \dot{\hat{x}} &= [A - LC]\hat{x} + [B - LD,\ L]u_c \\ y_c &= \hat{x} \end{aligned} \tag{4.5}$$

where $u_c = [u, y]^T$ is the combined input and y_c is the output. It is then decomposed into individual state equations for the purpose of implementation. For the sake of simplicity, the observer gain vector L is determined by placing the poles of the characteristic equation in one location.

$$\lambda(s) = \left| sI - (A - LC) \right| = (s + \omega_o)^3 \tag{4.6}$$

$$L = \begin{bmatrix} 3\omega_o, 3\omega_o^2, \omega_o^3 \end{bmatrix}^T$$

Discrete Estimator Design

The state space model in (4.3) is first discretized by applying Euler, ZOH, or FOH.

$$\begin{aligned}\hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) \\ \hat{y}(k) &= H\hat{x}(k) + J\,u(k)\end{aligned} \tag{4.7}$$

A discrete observer is created from this model.

$$\begin{aligned}\hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) + L_p(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= H\hat{x}(k) + J\,u(k)\end{aligned} \tag{4.8}$$

This is known as a predictive discrete estimator [40] because the current estimation error $y(k) - \hat{y}(k)$ is used to predict the next state estimate $\hat{x}(k+1)$. However, by defining the predictive estimator gain vector as

$$L_p = \Phi L_c, \tag{4.9}$$

the estimated state reduces to

$$\hat{x}(k+1) = \Phi\bar{x}(k) + \Gamma u(k) \tag{4.10}$$

where the new state includes a current measurement update, giving it less time delay.

$$\bar{x}(k) = \hat{x}(k) + L_c(y(k) - \hat{y}(k)) \tag{4.11}$$

This is referred to as a current discrete estimator [40]. When the sampling rate is low, this could play a significant role in enhancing the stability of a closed loop system. A block diagram from [63] is illustrated in Figure 13.

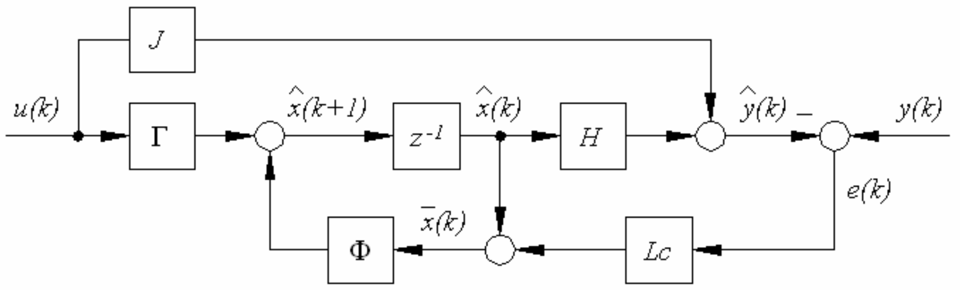


Figure 13: Current Discrete Estimator

The estimator is then rewritten to output the new state

$$\begin{aligned}\hat{x}(k+1) &= [\Phi - L_p H]\,\hat{x}(k) + [\Gamma - L_p J,\ L_p]\,u_d(k)\\ \bar{x}(k) &= [I - L_c H]\,\hat{x}(k) + [-L_c J,\ L_c]\,u_d(k)\end{aligned} \tag{4.12}$$

where $u_d(k)$ = [$u(k)$, $y(k)$]^T is the combined input and $\bar{x}(k)$ is the output. Similar to the Kalman filter, the first equation in (4.12) is the time update and the second equation is the measurement update. The only difference for the predictive estimator is that the output is just $\hat{x}(k)$.

For the sake of simplicity, the current estimator gain vector L_c is determined by placing the poles of the discrete characteristic equation in one location.

$$\lambda(z) = \left| zI - (\Phi - \Phi L_c H) \right| = (z-\beta)^3 \tag{4.13}$$

The relation between the discrete estimator poles and the continuous observer poles is given as

$$\beta = e^{-\omega_o T}\,. \tag{4.14}$$

In the past, numerical issues were associated with the continuous gain vector L because ω_o^n became very large. The discrete derivation should have better numerical precision because β is always between 0 and 1. The sample period T could be pulled out of the observer and combined in the controller such that every instance of ω_c is replaced with $\omega_c T$, which should always be a well conditioned value. Further research may lead to the establishment a standard initial value for $\omega_c T$, creating a starting point for ω_c and $\omega_o = k\omega_c$ to be determined prior to controller tuning.

Applying Euler to (4.3) and solving (4.13) for L_c yields

$$\Phi = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}, \quad L_c = \begin{bmatrix} 1-\beta^3 \\ (2-3\beta+\beta^3)\frac{1}{T} \\ (1-\beta)^3 \frac{1}{T^2} \end{bmatrix}. \qquad (4.15)$$
$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 \end{bmatrix}$$

where T is the discrete sample time. However, note that $[.]^T$ is denoted as the matrix transpose. In the past, the ESO was implemented by integrating each state equation in (4.4) using Euler [24], [51], [52], [58]-[61], [64]. The problem with this method is that it produces the same matrices as (4.15) except for $L_p = TL$, making the observer unstable at relatively low sample rates. Yet in cases where L is a nonlinear function, this may be the only way of discrete implementation. For the sake of further discussion, it is referred to as the Euler approximation.

Applying ZOH

$$\begin{aligned} \Phi &= e^{AT} \quad \Rightarrow \quad \sum_{k=0}^{\infty} \frac{A^k T^k}{(k)!} \\ \Gamma &= \int_0^T e^{A\tau} d\tau B \quad \Rightarrow \quad \sum_{k=0}^{\infty} \frac{A^k T^{k+1}}{(k+1)!} B \\ H &= C, \quad J = 0 \end{aligned} \qquad (4.16)$$

to (4.3) and solving (4.13) for L_c produces a more accurate estimation than Euler.

$$\Phi = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \ \Gamma = \begin{bmatrix} bT \\ b \\ 0 \end{bmatrix}, \ L_c = \begin{bmatrix} 1-\beta^3 \\ (1-\beta)^2(1+\beta)\frac{3}{2T} \\ (1-\beta)^3 \frac{1}{T^2} \end{bmatrix} \qquad (4.17)$$
$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \ J = \begin{bmatrix} 0 \end{bmatrix}$$

Note that it becomes an $\alpha - \beta - \gamma$ filter when b=0 [45].

Simulation and Analysis

The ESO is first applied in open loop to a simple plant

$$\ddot{y} = 50\dot{y} + 500u + 100w \tag{4.18}$$

where w is a 2.5*Hz* square wave starting at 0.3 sec. and u is a trapezoidal profile that lasts 0.125 sec. The estimator parameters are ω_o =300 and T = 0.005. A tracking error plot is shown in Figure 14 that compares the predictive and current discrete methods using both Euler and ZOH.

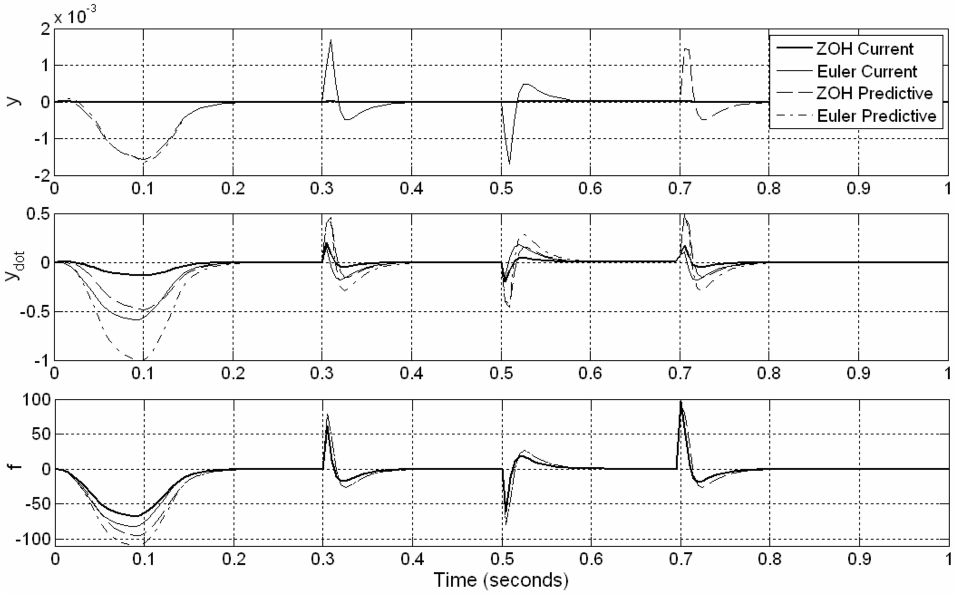


Figure 14: Open Loop Tracking Error

The transient and steady state parts of each trajectory are evaluated using integral absolute error and then summarized in the following table.

TABLE I: OPEN LOOP TRACKING ERRORS

Discretization Method	**Transient Integral Absolute Error**		
	y	*y'*	*f*
ZOH Current	1.49E-6	12E-3	5.90
Euler Current	1.51E-6	51E-3	7.16
ZOH Predictive	136E-6	41E-3	8.41
Euler Predictive	134E-6	87E-3	9.66
Discretization Method	**Steady State Integral Absolute Error**		
	y	*y'*	*f*
ZOH Current	0.10E-5	9E-3	2.83
Euler Current	0.10E-5	20E-3	2.83
ZOH Predictive	9.55E-5	30E-3	4.32
Euler Predictive	9.55E-5	40E-3	4.32

When the step size T = 0.005, the Euler approximation is unstable and therefore was not shown. However, the four methods shown use discrete pole placement and their outputs do not become unbounded until T = 0.066. From the table, the second most important option appears to be the current discrete method for tracking accuracy. The table also shows that ZOH is better than Euler and, interestingly, dominant in estimating transient velocity.

Next, the ESO is applied in closed loop to (4.18) and to a more complex simulation of an actual servo-motor.

$$\begin{aligned} &V_m = 80(75u - .075I_a), \quad |V_m| < 160, \quad |u| < 8 \\ &\dot{I}_a = 2500\ (V_m - .4I_a - 1.2\dot{y}) \\ &\ddot{y} = 11.1(100w + 1.5I_a) \end{aligned} \tag{4.19}$$

The closed loop method is outlined in Section 4.1.3. With ω_c =30 and ω_o =300, the sample period is increased to the point of instability and then tabulated below.

TABLE II: MAXIMUM CLOSED LOOP STEP SIZE

Discretization Method	Simple Plant (4.18)	Servo-motor (4.19)
Euler Approximation	26E-4	30E-4
Euler Predictive	37E-4	57E-4
Euler Current	47E-4	68E-4
ZOH Predictive	85E-4	140E-4
ZOH Current	150E-4	300E-4

The results show that the most important option for low sampling time requirements is ZOH, followed by the current discrete method. In this regard, the current discrete ESO with ZOH appears to be six to ten times better than the Euler approximation used in previous literature. The servo system in Section 4.1.3 was also simulated, resulting in an improvement of 5.3 times. In summary, the current discrete ESO with ZOH should be used for improved tracking accuracy as well as reduced delay.

4.1.2 Generalization of ESO

Although a second order example was used in the previous section, (4.4) through (4.14) and (4.16) are applicable to a plant of arbitrary order with any number of extended states. For example, a class of general n^{th} order plants similar to (4.1) is represented as

$$y^{(n)} = g(y, \cdots, y^{(n-1)}, t) + w + bu \tag{4.20}$$

where $y^{(n)}$ denotes the n^{th} derivative of the output and $g(y, \cdots, y^{(n-1)}, t)$ represents the internal dynamics. Two critical parameters are relative order n and high frequency gain b. Combining unknowns into one generalized disturbance $f(y, \cdots, y^{(n-1)}, w, t)$ results in

$$y^{(n)} = f(y, \cdots, y^{(n-1)}, w, t) + bu \; . \tag{4.21}$$

Note that when represented with an equivalent input disturbance $d = f / b$, the design model becomes

$$P_d(s) = b / s^n \; . \tag{4.22}$$

<u>Example 4.1</u>

From [65], consider the strictly proper linear time invariant plant.

$$P(s) = \frac{b_q s^q + b_{q-1} s^{q-1} + \cdots + b_0}{s^p + a_{p-1} s^{p-1} + \cdots + a_0} \tag{4.23}$$

By ignoring external disturbances, cross multiplying, integrating *p-q* times, and collecting terms before taking the inverse Laplace Transform, the system reduces to

$$y^{(p-q)} = f + b_q u \tag{4.24}$$

where

$$f = b_{q-1} \int u + b_{q-2} \iint u + \cdots + b_0 \overset{(q)}{\int} u - a_{p-1} \overset{(p-1)}{\int} y - a_{p-2} \overset{(p-2)}{\int} y - \cdots - a_0 y \tag{4.25}$$

and $\overset{(q)}{\int} u$ denotes the q^{th} integration of *u*. This example shows that, in actuality, the order $n = p - q$ equals the relative degree and the gain $b = b_q$ equals the high frequency gain. Therefore, (4.22) represents the high-frequency model or “most direct” path through the system. It will be used to determine controller parameters for a wide range of plants.

As a signal, the type of disturbance can be characterized similar that of system type in a classical control. This specification is outlined in [41] as the degree of a polynomial that approximates a signal, which directly relates to the number of times it is differentiated before reaching zero. Sometimes disturbances are represented by a set of cascaded integrators $1/s^h$ with unknown input [40], [42], [47]-[49], [55]. Under this assumption, the plant is represented in Figure 15 by two sets of cascaded integrators; one for the design model and another for the disturbance model.

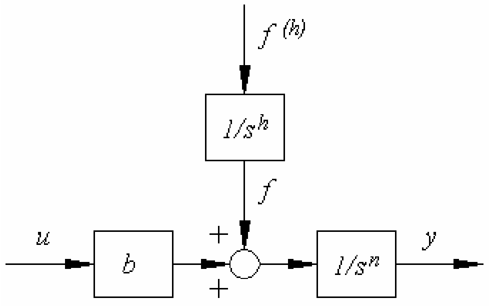


Figure 15: Canonical Form with Disturbance Model

It will also be shown that this assumption leads to an estimated disturbance equivalent to that of a DOB.

In (4.3), previous ESO design [24], [51], [52], [58]-[61], [64] and in [40], [54], [55] disturbances are considered to be piece-wise constant with $h = 1$ or a series of steps. Now an ESO with $h = 1, 2, 3$ can respectively track a square, triangular, or parabolic disturbance. A sinusoid is a different matter because it is infinitely differentiable. However, increasing h increases the degree of the polynomial and improves tracking of a sinusoid or any time varying disturbance. An ESO with h extended states for a relative n^{th} order plant is denoted as an $\text{ESO}_{n,\,h}$.

Continuous Observer Design

The new form is represented in state space

$$\begin{aligned}\dot{x} &= Ax + Bu + Ef^{(h)}\\ y &= Cx + Du\end{aligned} \tag{4.26}$$

where the state includes the disturbance *f* and its derivatives to be estimated.

$$\begin{aligned}x &= [x_1, \cdots, x_n, x_{n+1}, \cdots, x_{n+h}]^T\\ &= [y^{(0)}, \cdots, y^{(n-1)}, f^{(0)}, \cdots, f^{(h-1)}]^T\end{aligned} \tag{4.27}$$

Since the new form consists of cascaded integrators, the *A* matrix simplifies to an *n+h* square matrix with ones on the super diagonal. Each element of *A* is defined as

$$a_{i,j} = \begin{cases}1, i = j-1\\ 0, otherwise\end{cases}. \tag{4.28}$$

Since the input is added after *n*-integrators, the first state is defined as the output, and the derivative of the last state is $f^{(h)}$, the other matrices become

$$B = \begin{bmatrix}0_{n-1}\\ b\\ 0_h\end{bmatrix},\quad C = \begin{bmatrix}1 & 0^T_{n+h-1}\end{bmatrix},\quad D = 0,\quad E = \begin{bmatrix}0_{n+h-1}\\ 1\end{bmatrix} \tag{4.29}$$

where 0_h represents a $h \times 1$ zero vector.

For the sake of simplicity, the observer gain vector is determined by placing all of the poles of the characteristic equation in one location.

$$\lambda(s) = \left| sI - (A - LC) \right| = (s + \omega_o)^{n+h} \tag{4.30}$$

As a result, each element in *L* becomes

$$l_i = c_{n+h,\,i}\,\omega_o^i,\quad i = 1,2,\cdots,n+h \tag{4.31}$$

where the binomial coefficients are $c_{i,\,j} = \begin{pmatrix}i\\ j\end{pmatrix} = \dfrac{i!}{j!(i-j)!}$.

The ESO can also be represented in filter form

$$\begin{aligned} \hat{y}^{(i)} &= s^i[Q_y y + (1-Q_y)P_d u], \quad i = 0,\cdots,n-1 \\ \hat{f}^{(j)} &= s^j[bQ_f(P_d^{-1}y-u)], \quad j = 0,\cdots,h-1 \end{aligned} \tag{4.32}$$

where binomial filters

$$Q_y(s) = \frac{\beta_{n+h,\,n+h-i-1}(s)}{\beta_{n+h,\,n+h}(s)}, \quad Q_f(s) = \frac{\beta_{n+h,\,h-j-1}(s)}{\beta_{n+h,\,n+h}(s)} \tag{4.33}$$

consist of numerator and denominator polynomials that are functions of a single tuning parameter $\omega_o = 1/\tau$.

$$\beta_{i,j}(s) = 1 + \sum_{r=1}^{j} c_{ir}(\tau s)^r \tag{4.34}$$

This form shows that additional extended states raise observer order, *n*+*h*, and increase the slope at the cutoff frequency. It also shows that the estimated disturbance is equivalent to a DOB, i.e. a filtered version of the actual *f*.

$$\hat{f} = bQ_f(P_d^{-1}y-u) \tag{4.35}$$

<u>*Discrete Estimator Design*</u>

Applying ZOH to (4.26) using (4.16) produces an *n*+*h* square Φ matrix where each element is defined as

$$\phi_{i,j} = \begin{cases} \gamma_{j-i}, & i \le j \\ 0, & otherwise \end{cases} \tag{4.36}$$

for $\gamma_k = T^k / k!$. The Γ matrix reduces to

$$\Gamma = \begin{bmatrix} b\gamma_n & ,\cdots, & b\gamma_1, & 0_h \end{bmatrix}^T . \tag{4.37}$$

If FOH is preferred, the only change is in the Γ and *J* matrices, which become

$$\begin{aligned} \Gamma &= \left[\frac{(2^{n+1}-2)}{n+1} b\gamma_n \quad ,\cdots, \quad \frac{2}{2} b\gamma_1, \quad 0_h \right]^T \\ J &= \left[\frac{b\gamma_n}{n+1} \quad ,\cdots, \quad \frac{b\gamma_1}{2}, \quad 0_h \right]^T \end{aligned} . \tag{4.38}$$

For the sake of simplicity, the current estimator gain vector L_c is determined by placing the poles of the discrete characteristic equation in one location.

$$\lambda(z) = \left| zI - (\Phi - \Phi L_c H) \right| = (z-\beta)^{n+h} \tag{4.39}$$

As a result, the current estimator gain vector is listed in the table as a function of *n+h*. A current discrete ESO with *h* extended states for a relative n^{th} order plant is denoted as a CDESO$_{n,\,h}$.

TABLE III: CDESO Estimator Gains for ZOH and FOH

n+h	L_c
1	$[1-\beta]^T$
2	$[1-\beta^2, (1-\beta)^2 \frac{1}{T}]^T$
3	$[1-\beta]^T [1-\beta^3, (1-\beta)^2(1+\beta)\frac{3}{2T}, (1-\beta)^3 \frac{1}{T^2}]^T$
4	$[1-\beta^4, (1-\beta)^2(11+\beta(14+11\beta))\frac{1}{6T},$ $(1-\beta)^3(1+\beta)\frac{2}{T^2}, (1-\beta)^4 \frac{1}{T^3}]^T$
5	$[1-\beta^5, (1-\beta)^2(1+\beta)(5+\beta(2+5\beta))\frac{5}{12T},$ $(1-\beta)^3(7+\beta(10+7\beta))\frac{5}{12T^2},$ $(1-\beta)^4(1+\beta)\frac{5}{2T^3}, (1-\beta)^5 \frac{1}{T^4}]^T$

4.1.3 Design Example

A simulation of an industrial motion control test bed is used to demonstrate the control design procedure and its simplicity, resulting performance, and overall effectiveness in the absence of a simulation model. The servo amplifier, motor, and drive train are modeled with a resonant load as

$$\begin{aligned} V_m &= 4(V_c - 2.05 I_a), \quad |V_c| < 4.5, \quad |V_m| < 10 \\ \dot{I}_a &= 2500(V_m - 4I_a - .2\dot{x}_m), \quad |I_a| < 1 \\ T_m &= .5I_a - T_d - T_l \\ T_l &= .0005(\dot{x}_m - 4\dot{x}_l) + .0001(x_m - 4x_l) \\ \ddot{x}_m &= 2500 T_m \\ \ddot{x}_l &= 175 T_l \end{aligned} \tag{4.40}$$

where V_c, x_l, and T_d are the control input voltage, output load position, and torque disturbance, respectively. Backlash of a ±0.31μm/sec. dead-bandwidth on $\dot{x}$ is also applied.

The control design method using the ESO is fairly straight forward with only a few physical intuitions. In the most basic sense, a servo motor can be considered as a double integrator.

$$\frac{x_l(s)}{V_c(s)} \approx \frac{b_m}{s^2}\,. \tag{4.41}$$

It is put into the new canonical form where $f(t)$ represents any of the discrepancies or dynamics not modeled in (4.41).

$$\ddot{x}_l(t) = f(t) + b_m V_c(t) \tag{4.42}$$

First, a CDESO$_{2,h}$ is used to estimate $x_l(t)$, $\dot{x}_l(t)$, and $f(t)$ in discrete time. Then the estimated disturbance is fed back to cancel itself

$$V_c(k) = \frac{u_0(k) - \hat{f}(k)}{b_m} \tag{4.43}$$

which reduces the system to a double integrator.

$$\ddot{x}_l(t) \approx u_0(t) \tag{4.44}$$

Finally, a parameterized control law is used to control the augmented system where *r*(*k*) is a reference motion profile.

$$u_0(k) = \omega_c^2(r(k) - \hat{x}_l(k)) - 2\omega_c \hat{\dot{x}}_l(k) \tag{4.45}$$

The observer and the control laws in (4.43) and (4.45) are selected with a sample rate of 10 kHz to control the motion system's model in (4.40). The gain $b_m = 25$ is crudely estimated as the initial acceleration from a step response. Disturbance rejection was tested by applying various torque disturbances at time t = 1 second and 0.1% white noise is injected into the output.

Keeping the control signal within ±4.5V and its noise level within ±100mV, ω_c and ω_o were increased to 50 and 150, respectively. The results for a type 1 square torque disturbance are shown in Figure 16.

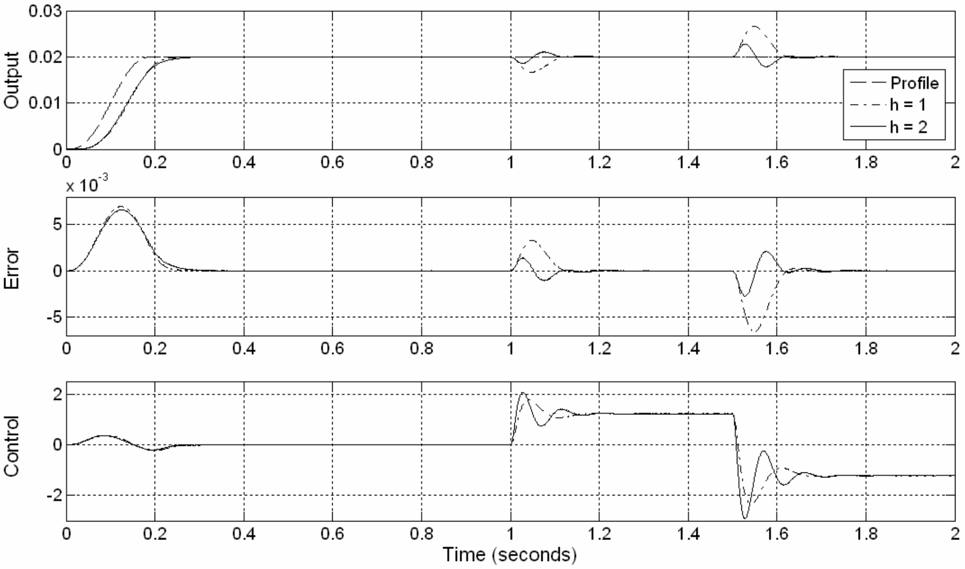


Figure 16: Response to a Square Torque Disturbance

Robustness was tested by increasing the load by a factor of nearly 8. There was no noticeable difference. The results for a type 2 triangular and type ∞ sinusoidal torque disturbance are shown in Figure 17 and Figure 18, respectively.

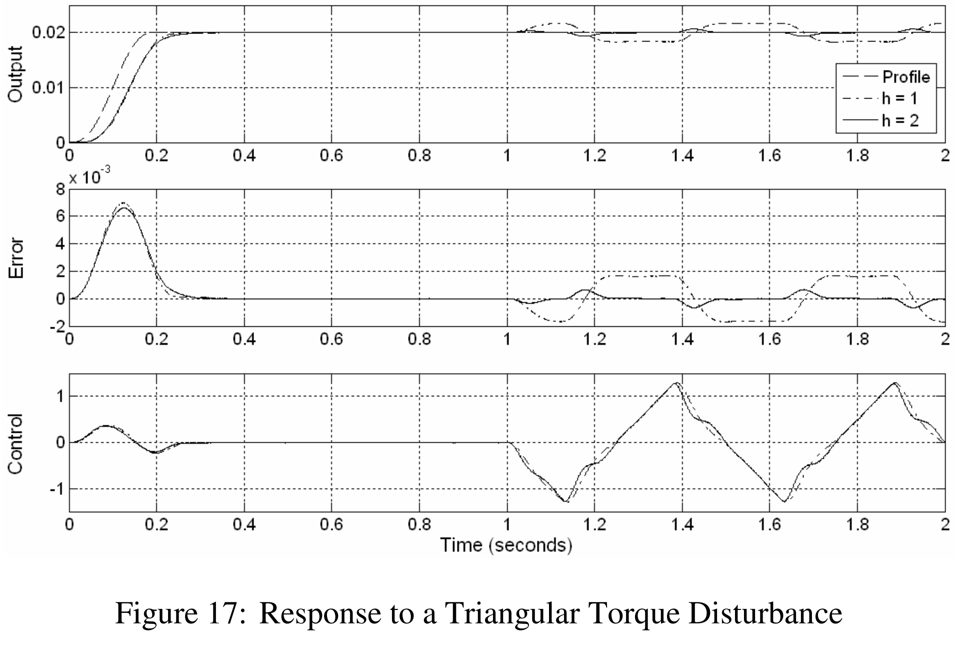


Figure 17: Response to a Triangular Torque Disturbance

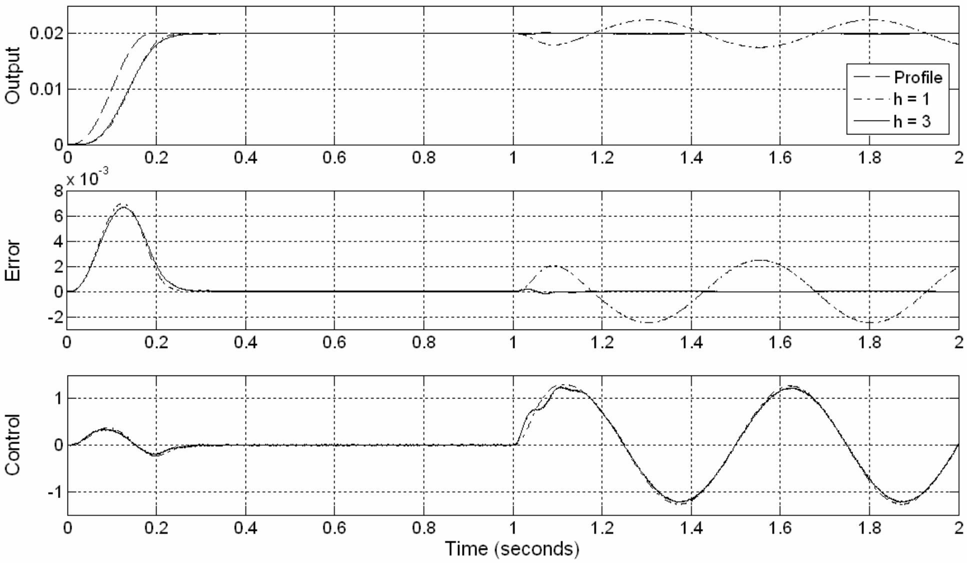


Figure 18: Response to a Sinusoidal Torque Disturbance

The results show that two extended states reduce the error compared to one extended state. In Fig. 17, two extended states drive to zero the error created by type 2 disturbances. Although a sinusoidal disturbance is infinitely differentiable, three extended states significantly reduce the steady state error in Fig. 18.

Remarks

Various discrete implementations of the extended state observer were studied and compared. It is shown that the current discrete formulation is superior to the predictive one in reducing the delay associated with the sampling process. It is also demonstrated that the ZOH implementation improves estimation accuracy without additional complexity to the user. To facilitate the ESO implementation for practitioners, the algorithm is derived symbolically with a single tuning parameter, i.e. the bandwidth of the observer. Another significant development is the generalization of the ESO for various types of systems and disturbances. Finally, a filter version shows that the estimated disturbance is equivalent to the DOB structure. Unlike the DOB, however, the ESO estimates suitable derivatives of the output that are used by the controller.

4.2 Generalization of PID

The PID control law is generalized to accommodate higher order systems. A prefilter is proposed as a function of the new control law to eliminate the need for independent prefilter design. This forms a unique 2-DOF structure that is then parameterized as a function of closed loop bandwidth by representing the plant in the form of (4.21). The result is a powerful controller with a transparent design procedure.

The Section is organized as follows. Section 4.2.1 introduces a GPID framework. Controller tuning and a design procedure are given in Sections 4.2.2 and 4.2.3, respectively, and stability and an alternative to output differentiation are proposed in Sections 4.2.4 and 4.2.5. Section 4.2.6 demonstrates its effectiveness in hardware on a motion control platform and Section 4.2.7 proposes the combination of ADRC and GPID.

4.2.1 Extending the Control Law

The traditional PID controller is defined as a transfer function.

$$C(s) = k_i s^{-1} + k_p s^0 + k_d s^1 \tag{4.46}$$

A generalization of this controller is proposed as follows.

$$C(s) = k_{-m}s^{-m} + \cdots + k_{-1}s^{-1} + k_0 s^0 + k_1 s^1 + \cdots + k_n s^n \tag{4.47}$$

As a rule of thumb, plants lacking a pole at the origin use integral control, second order dominant plants use derivative control, and all plants typically use proportional control. Extending this logic to n^{th} order plants produces

$$C(s) = k_{-m}s^{-m} + k_{1-m}s^{1-m} + \cdots + k_{n-1}s^{n-1} \tag{4.48}$$

where m is equal to the number of pure integrators or extended states in the controller and $1\text{-}m$ is less than or equal to the number of pure integrations in the plant.

Expansion to a 2-DOF Structure

The traditional PID control structure in Figure 19 is shown with controller C and plant P.

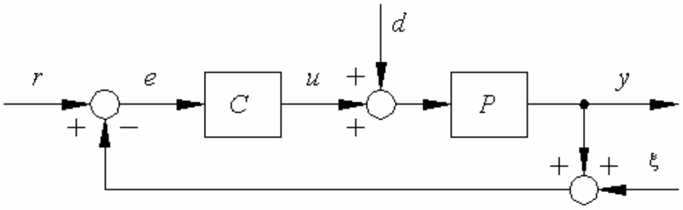


Figure 19: Unity Feedback Control Structure

Closed loop response and disturbance rejection are defined by the following transfer functions, respectively.

$$G_{ry} = \frac{CP}{1+CP}, \quad G_{dy} = \frac{P}{1+CP} \tag{4.49}$$

Clearly the controller must be tuned differently for each of the above transfer functions to achieve the best performance. Next, consider a 2-DOF configuration in Figure 20.

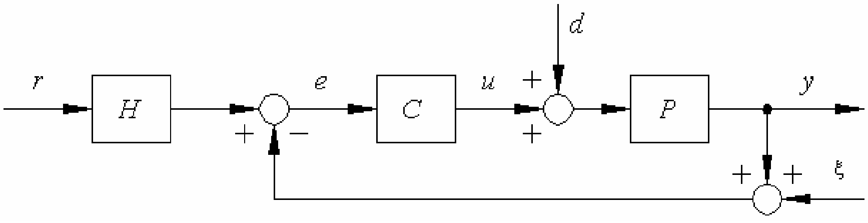


Figure 20: A 2-DOF Control Structure

It has the capacity to be highly robust when command following is uniquely determined by *H* after disturbance rejection is determined by *C* [32], [50], [53], [54], [57], [66].

$$G_{ry} = \frac{HCP}{1+CP}, \quad G_{dy} = \frac{P}{1+CP} \tag{4.50}$$

To gain the benefit of the added complexity while retaining design simplicity, an equivalent configuration is shown in Figure 21 with $C=C_1+C_2$ and $H=C_1C^{-1}$.

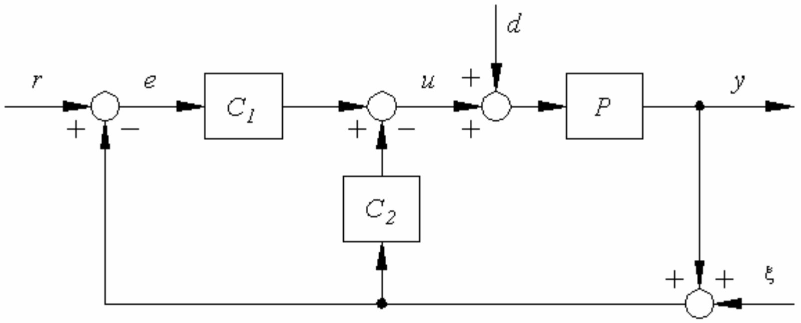


Figure 21: An Equivalent 2-DOF Control Structure

A 2-DOF control law emerges by separating the first *h+1* terms from (4.48).

$$\begin{aligned} C_1(s) &= k_{-m}s^{-m} + \cdots + k_{h-m}s^{h-m} \\ C_2(s) &= k_{h-m+1}s^{h-m+1} + \cdots + k_{n-1}s^{n-1} \end{aligned} \tag{4.51}$$

Here, *h* is equal to the number of zeros in the closed loop transfer function that are generated by the controller. When *h=m+n-1*, the structure reduces to 1-DOF with $C_1=C$ and $C_2=0$.

When *h=0*, the new structure represents a set of controllers that do not generate additional zeros in the closed loop transfer function. This provides a simpler design for most applications.

$$C_1(s) = k_{-m}s^{-m}, \quad C_2(s) = k_{1-m}s^{1-m} + \cdots + k_{n-1}s^{n-1} \tag{4.52}$$

Disturbances are often modeled using differential equations. Sometimes they are represented by a set of cascaded integrators $1/s^m$ and have a fictitious zero input [47], [54], [67]. Under this assumption, the Internal Model Principle specifies that the controller should also include a factor of $1/s^m$ in order to reject such disturbances [68]. When disturbances are assumed to be piece-wise constant over a given period of time, meaning *m=1*, the control law is simply extended by one state and yet the controller exhibits a relatively high degree of robustness. For example, the Integral Control structure is used in this way to eliminate steady state errors by combining state feedback with the integration of the error [40], [55].

When *m=1*, the new structure becomes a special case of integral control, since state feedback is strictly composed of the output and its derivatives.

$$C_1(s) = k_{-1}s^{-1}, \quad C_2(s) = k_0 s^0 + \cdots + k_{n-1}s^{n-1} \tag{4.53}$$

When *m* is positive, the new structure is a generalization of this special case. It also represents a generalization of the I-PD controller that is common to industry.

4.2.2 Controller Tuning

Using the model from (4.22) and the control law from (4.51), the nominal closed loop transfer function is derived.

$$G_{ry}(s) = \frac{k_{h-m}s^h + \cdots + bk_{1-m}s + bk_{-m}}{s^{n+m} + bk_{n-1}s^{n+m-1} + \cdots + bk_{1-m}s + bk_{-m}} \tag{4.54}$$

Controller gains are determined by setting it equal to a desired transfer function having the same general form and a single tuning parameter $\omega_c = 1/\tau$ with a concrete fundamental interpretation [50], [52], [53], [54], [56], [57], [67], [69], [70].

$$G^*_{ry}(s) = \frac{\beta_{n+m,\,h}(s)}{\beta_{n+m,\,n+m}(s)}, \quad \beta_{ij}(s) = 1 + \sum_{r=1}^{j} c_{ir}(\tau s)^r \tag{4.55}$$

When all of the poles are placed in one location to simplify design, (4.55) reduces to the commonly used binomial $Q_{n+m,h}$ filter. The binomial coefficients $c_{ir} = \begin{pmatrix} i \\ r \end{pmatrix} = \dfrac{i!}{r!(i-r)!}$ then yield a closed form solution for the controller gains.

$$k_i = \begin{pmatrix} n+m \\ n-i \end{pmatrix} \frac{\omega_c^{n-i}}{b}, \quad i = -m, 1-m, \cdots, n-1 \tag{4.56}$$

<u>Example 4.2</u>

When *h=0*, the nominal closed loop transfer function is derived using the control law from (4.51).

$$G_{ry}(s) = \frac{bk_{-m}}{s^{n+m} + bk_{n-1}s^{n+m-1} + \cdots + bk_{1-m}s + bk_{-m}} \tag{4.57}$$

By selecting the gains according to (4.56), the transfer function reduces to a $Q_{n+m,0}$ filter. Disturbance rejection is also characterized by a filter.

$$G_{ry}(s) = \frac{\omega_c^{n+m}}{(s+\omega_c)^{n+m}}, \quad G_{dy}(s) = \frac{bs^m}{(s+\omega_c)^{n+m}}. \tag{4.58}$$

Setting *m=1* is sufficient for most applications. Although additional extended states do improve the disturbance rejection property of the controller and increase the slope at the cutoff frequency, they also introduce more phase lag. As a result, compensation for phase lag is addressed in the next section by increasing the bandwidth.

The table below gives examples of controllers with *h=0* and *m=1* for first through third order plants. In most cases, this is all that is required for good performance.

TABLE IV: AN EXAMPLE OF GPID CONTROLLERS

n	*C(s)*	*C₁(s)*	*C₂(s)*	*Gains*
1	PI	$k_i s^{-1}$	k_p	$k_i = \omega_c^2 / b$ $k_p = 2\omega_c / b$
2	PID	$k_i s^{-1}$	$k_p + k_d s$	$k_i = \omega_c^3 / b$ $k_p = 3\omega_c^2 / b$ $k_d = 3\omega_c / b$
3	PIDD	$k_i s^{-1}$	$k_p + k_d s + k_2 s^2$	$k_i = \omega_c^4 / b$ $k_p = 4\omega_c^3 / b$ $k_d = 6\omega_c^2 / b$ $k_2 = 4\omega_c / b$

Similar to increasing the number of extended states *h* in the ESO, the number of extended states *m* in the GPID controller can be increased to provide rejection to different types of disturbances.

4.2.3 Design Procedure

Preliminary system identification is often unnecessary because the only design parameters are ω_c and b and they have a direct impact on the bandwidth and overshoot of the output, meaning they can easily be adjusted by the user. In practice, *1/b* needs to be initially within fifty percent of its true value, the total inertia in a second order system, and such a broad range is frequently known.

A step is usually not used as a profile for PID controllers because the abrupt change is directly translated to the control signal through the proportional gain and amplified further through the derivative gain. However, the new structure does not have this problem because the input is always filtered by integrators when $h=0$ and $m>0$. The result is a smoother control signal in a smaller range.

When a step input is applied to the system in the previous example, normalized settling times for $\omega_c=1$ are tabulated below.

TABLE V: NORMALIZED SETTLING TIMES FOR VARIOUS ORDERS

$n+m$	1	2	3	4	5
t_n	3.9124	5.8342	7.5169	9.0842	10.5807

Since a step is the fastest possible profile, the minimum settling time of the system for a given bandwidth then becomes

$$t_s = t_n / \omega_c . \tag{4.59}$$

When faced with time specifications, this can serve as a starting point for tuning ω_c or it can determine if a solution is even feasible.

When a profile with settling time t_p is used, the total settling time of the system is then approximated by

$$t_t \approx t_p + t_s \,. \tag{4.60}$$

A design and tuning procedure is now given. The idea is to set the controller bandwidth as high as possible. A profile is then typically used to achieve a slower settling time or to meet control signal constraints, but a step is used if the fastest possible response is desired without regard to exact trajectory.

1. Select an appropriate controller from (4.52) with $m>0$.

2. If required, design any approximate derivative filters with bandwidth $\omega_o=10\omega_c$, so they do not interfere with the phase at the crossover frequency.

3. Set ω_c using (4.59) according to initial specifications.

4. If required, design a transient profile using (4.60) according to specifications or to avoid control signal saturation. A rule of thumb $t_p=1\sim5t_s$ can also be used.

5. Run the closed loop system. If required, adjust b to a point just before overshoot appears in the output.

6. Increase ω_c and ω_o together to a point just before oscillation appears in the control signal.

7. Adjust the relation between ω_c and ω_o to meet the design specifications for noise level and disturbance rejection.

8. Readjust the profile time to meet settling time specifications or to avoid control signal saturation.

Example 4.3

Consider a process that is difficult to control with PID; one with three equal lags.

$$P(s) = \frac{1000}{(s+a)^3} e^{-.001s} \tag{4.61}$$

First, a controller is selected from Table IV for $n = 3$.

$$u = (\omega_c^4 \int (r-y)dt - 4\omega_c^3 y - 6\omega_c^2 \dot{y} - 4\omega_c \ddot{y}) / b \tag{4.62}$$

Approximate single and double differentiators are applied.

$$\dot{y} \approx \frac{s\omega_o^2}{(s+\omega_o)^2} y, \quad \ddot{y} \approx \frac{s^2\omega_o^3}{(s+\omega_o)^3} y \tag{4.63}$$

Initially, $\omega_o = 10\omega_c$ and $\omega_c = 9.1$, which corresponds to a settling time of one second for a fourth order step response. The system was induced with 0.1% measurement white noise. Then ω_c and ω_o were increased to 20 and 350, respectively, keeping the output smooth and the control signal noise level within ±100mV.

In order to test for disturbance rejection, a step disturbance corresponding to 50% of the maximum control signal is added at t = 2 seconds. In order to test for robustness, the location of the poles is varied by a factor of 100 in successive simulations without retuning the controller. Simulation results to a step input are illustrated in Figure 22.

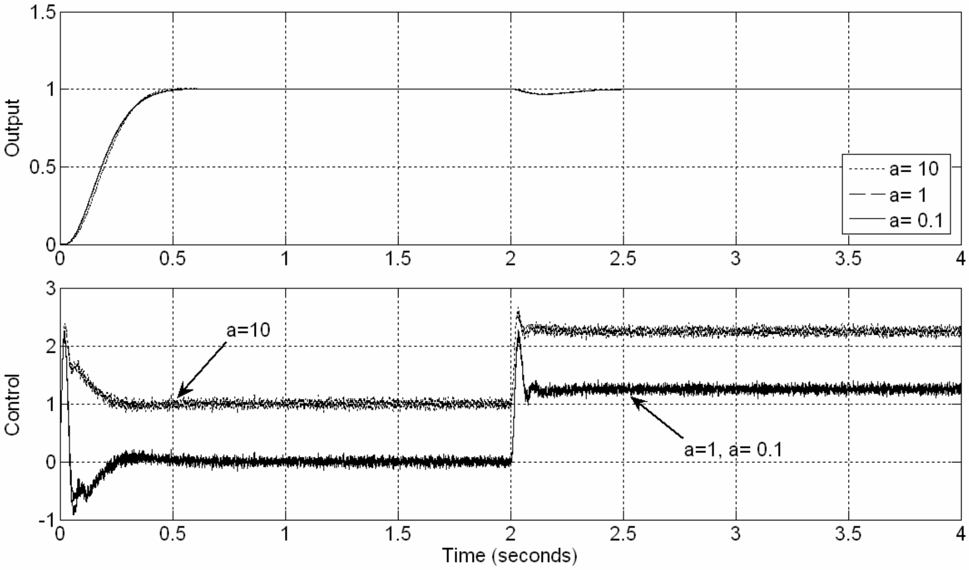


Figure 22: Three Lag System with GPID

4.2.4 Stability of LTI Systems

In this section, the parametric stability bounds are presented for specific forms of first through third order systems. Since PID and GPID have the same loop gain $L = CP$, both structures have identical stability and integrator windup properties. For the sake of simplicity, the $m = 1$ case is analyzed.

Consider strictly proper LTI plants from Example 4.1 of the form

$$P(s) = \frac{b_q s^q + b_{q-1} s^{q-1} + \cdots + b_0}{s^p + a_{p-1} s^{p-1} + \cdots + a_0} \tag{4.64}$$

where $n = p - q$. When the control law in (4.53) is applied to (4.64), the characteristic equation of the closed loop system becomes

$$\lambda(s) = s^{p+1} + d_p s^p + \cdots + d_0 \tag{4.65}$$

where each coefficient is defined by

$$d_i = b_i k_{-1} + \cdots + b_0 k_{i-1} + a_{i-1}, \quad i = 0, \cdots, p. \qquad (4.66)$$

Nonexistent plant coefficients and controller gains generated in (4.66) are ignored.

A necessary condition for the stability of an LTI system is that all coefficients of $\lambda(s)$ are non-negative. Necessary and sufficient conditions are generated from the Routh-Hurwitz criterion [71], [72], resulting in a set of inequalities that place n-dimensional limits on plant coefficients, given the controller gains $k_j,\ j = -1, \cdots n-1$.

Assuming all controller gains are positive, the stability condition for first order plants in (4.64) is determined by a single inequality, $d_1 > 0$. For second order plants, the condition $d_2 > d_0 / d_1$ creates a curve, forming the boundary of a stability region to include all stable, as well as some unstable plant coefficients. For third order plants, two conditions, $d_3 > d_1 / d_2$ and $d_2 > d_1 / d_3 + d_0 d_3 / d_1$, form a boundary surface. The first inequality is used to partition the surface generated by the second inequality, the more restrictive of the two conditions.

Example 4.4

PID is widely used in industrial motion control systems. Oftentimes plants are representative of the form

$$P(s) = \frac{b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_0}. \qquad (4.67)$$

For example, a set of models of this type is used in [32] as test cases for various PID algorithms. The control law in (4.53) is applied to (4.67) which forms a characteristic equation with coefficients that are defined by

$$d_i = b_0 k_{i-1} + a_{i-1}, \quad i = 0, \cdots, n. \tag{4.68}$$

The results are tabulated below.

TABLE VI: STABILITY BOUNDS UP TO THIRD ORDER

n	Boundary	Type
1	$a_0 > -b_0 k_0$	interval
2	$a_1 > \dfrac{b_0 k_{-1}}{b_0 k_0 + a_0} - b_0 k_1$	curve
3	$a_2 > \dfrac{b_0 k_0 + a_0}{b_0 k_1 + a_1} - b_0 k_2$ $a_1 > \dfrac{b_0 k_{-1}(b_0 k_2 + a_0)}{b_0 k_0 + a_0} + \dfrac{b_0 k_0 + a_0}{b_0 k_2 + a_2} - b_0 k_0$	surface

Controller parameters are determined by placing all of the poles of $\lambda(s)$ in one location where $\bar{b}_0, \bar{a}_j$ are estimates of plant coefficients.

$$k_j = \binom{n+1}{n-j} \frac{\omega_c^{n-j} - \bar{a}_j}{\bar{b}_0}, \quad j = -1, \cdots, n-1 \tag{4.69}$$

In most cases $\bar{a}_j$ are unknown and therefore set to zero, as is the case for the parameterization in (4.56). However, when they are accounted for, i.e. model based control, the coefficients of $\lambda(s)$ become

$$d_{i+1} = \Delta b \binom{n+1}{n-i} \omega_c^{n-i} - \Delta b \bar{a}_i + a_i, \quad i = -1, \cdots, n-1 \tag{4.70}$$

where the multiplicative perturbation $\Delta b = b_0 / \bar{b}_0$ represents how precise the gain estimate is with respect to the actual gain. As a convention, ω_c and Δb are positive in all cases. The stability bounds as a result of GPID parameterization for first through third order systems are tabulated below.

TABLE VII: PARAMETERIZED STABILITY BOUNDS UP TO THIRD ORDER

n	Boundary	Type
1	$a_0 > -\Delta b(2\omega_c - \bar{a}_0)$	interval
2	$a_1 > \dfrac{\Delta b\omega_c^3}{\Delta b(3\omega_c^2 - \bar{a}_0) + a_0} - \Delta b(3\omega_c - \bar{a}_1)$	curve
3	$a_2 > \dfrac{\Delta b(4\omega_c^3 - \bar{a}_0) + a_0}{\Delta b(6\omega_c^2 - \bar{a}_1) + a_1} - \Delta b(4\omega_c - \bar{a}_2)$ $a_1 > \dfrac{\Delta b(4\omega_c^3 - \bar{a}_0) + a_0}{\Delta b(4\omega_c - \bar{a}_2) + a_2} + \dfrac{\Delta b\omega_c^4(\Delta b(4\omega_c - \bar{a}_2) + a_2)}{\Delta b(4\omega_c^3 - \bar{a}_0) + a_0} - \Delta b(6\omega_c^2 - \bar{a}_1)$	surface

Figure 23 shows how the region for a second order plant expands proportional to the bandwidth of the plant when the controller bandwidth is increased and $\bar{a}_j$ set to zero.

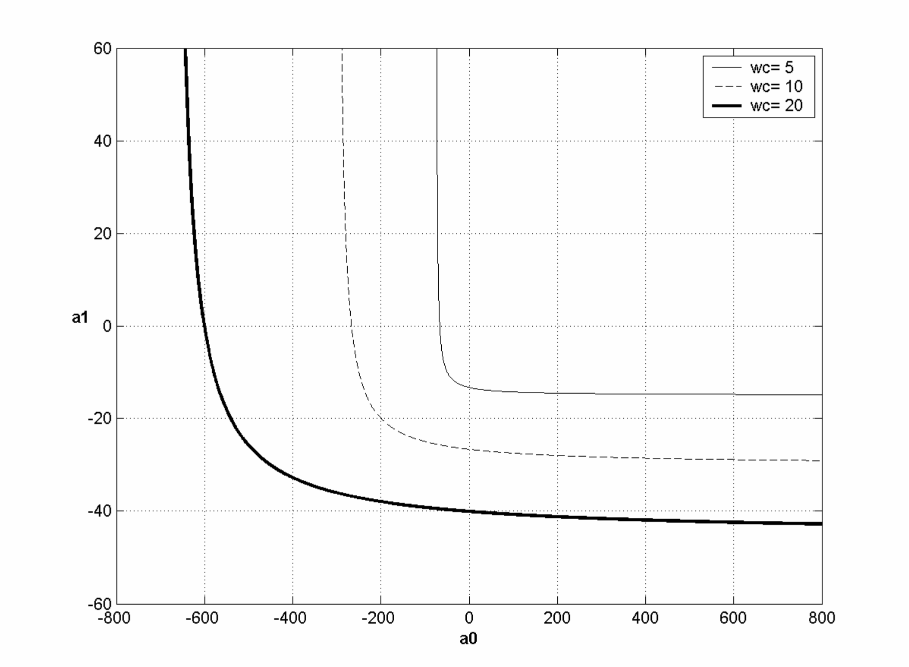


Figure 23: Stability Bound for Second Order Plants

Figure 24 shows an example of a surface for a third order plant.

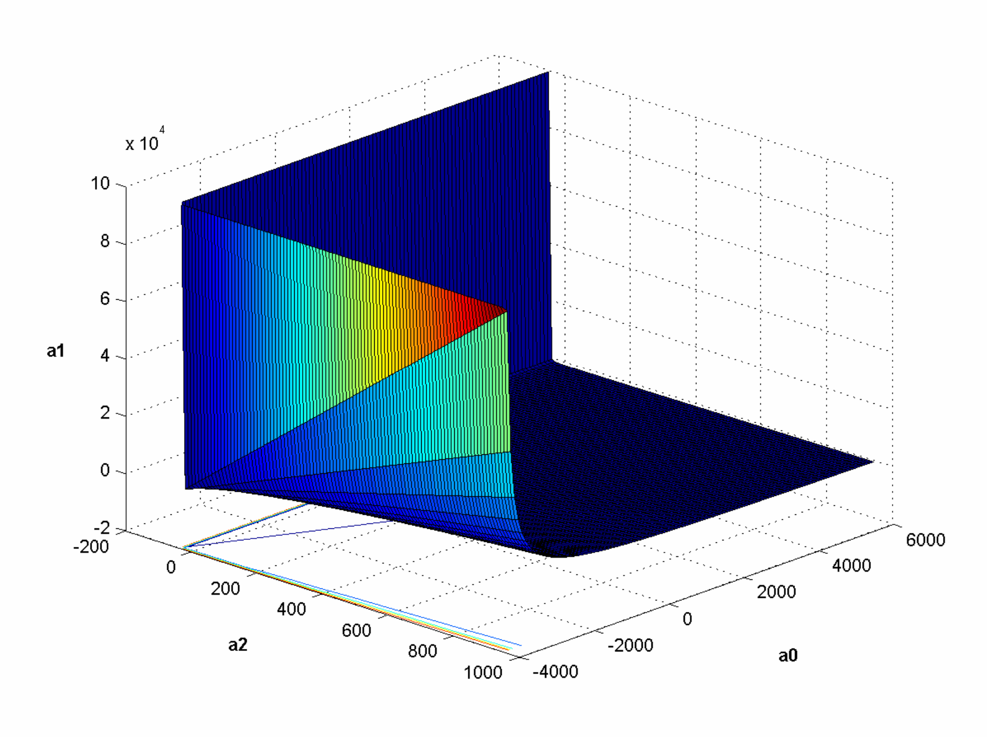


Figure 24: Stability Bound for Third Order Plants

Robustness

It is clear that the first term in (4.70) is dominant. To give insight into how sensitive the controller is to changes in model parameters, $\lambda(s)$ is rewritten as

$$\lambda(s) = s^{n+1} + \sum_{j=1}^{n} c_{n+1,\,j}\xi_j \omega_c^{n-j+1} s^j + \Delta b \omega_c^{n+1} \tag{4.71}$$

where the additive perturbations are $\Delta a_j = a_j - \overline{a}_j$ and the damping ratios are defined as

$$\xi_j = \Delta b + \frac{\Delta a_j}{c_{n+1,\,j}\omega_c^{n-j+1}} \tag{4.72}$$

This formulation shows that model uncertainty mostly affects the damping ratios. The most sensitive parameter $\bar{b}_0$ has to be miscalculated by factors of b_0 before it begins to affect overshoot. This strong correlation to damping allows it to be tuned easily; i.e. increased until the response shows signs of overshoot and then decreased until the overshoot nearly disappears. Additionally, $\bar{b}_0$ acts as a scaling factor to reduce $G(s)$ to unity gain. Coefficients a_j must be miscalculated by factors of $c_{n+1,\,j}\omega_c^{n-j+1}$ before they begin to affect damping, i.e. they become insignificant when $c_{n+1,\,j}\omega_c^{n-j+1}$is large, even when all $\bar{a}_j$ are set to zero.

4.2.5 Output Differentiation

With GPID, a robust controller is designed to reject the generalized disturbance without modeling it, meaning only the design model P_d is considered in the design. Since P_d in (4.22) consists of cascaded integrators, a state observer can be constructed and used as an alternative form of differentiating the output. By defining the states as $x = [x_1, x_2, \cdots, x_n]^T = [y, \dot{y}, \cdots, y^{(n-1)}]^T$, a state space model of P_d is created.

$$\begin{aligned}
&\dot{x} = Ax + Bu \\
&y = Cx \\
&A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b \end{bmatrix} \\
&C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}
\end{aligned} \tag{4.73}$$

An observer of this model is constructed with a gain vector $L = [l_1, \cdots, l_n]^T$.

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \tag{4.74}$$

Since (4.52) is already in state feedback form, the last n terms in C_2 are modified to accept the observer's output.

$$C_2 = k_{1-m} \overset{(m-1)}{\int} y - \cdots - k_{-1} \int y - k_0 \hat{x}_1 - \cdots - k_{n-1} \hat{x}_n \tag{4.75}$$

For the sake of simplicity, all of the poles of the observer's characteristic equation are placed in one location

$$\lambda(s) = | sI - (A - LC) | = (s + \omega_o)^n \tag{4.76}$$

As a result, the observer gains become

$$l_i = \begin{pmatrix} n \\ i \end{pmatrix} \omega_o^i, \quad i = 1, 2, \cdots, n \tag{4.77}$$

<u>*Example 4.5*</u>

A controller for example 4.3 becomes

$$u = (\omega_c^4 \int (r - y) dt - 4\omega_c^3 \hat{x}_1 - 6\omega_c^2 \hat{x}_2 - 4\omega_c \hat{x}_3) / b \tag{4.78}$$

with the following observer.

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + 3\omega_o (y - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + 3\omega_o^2 (y - \hat{x}_1) \\ \dot{\hat{x}}_3 = bu + \omega_o^3 (y - \hat{x}_1) \end{cases} \tag{4.79}$$

Figure 25 shows how a more accurate differentiation signal allows the bandwidth to be increased while keeping the output smooth and the control signal noise level within ±100mV.

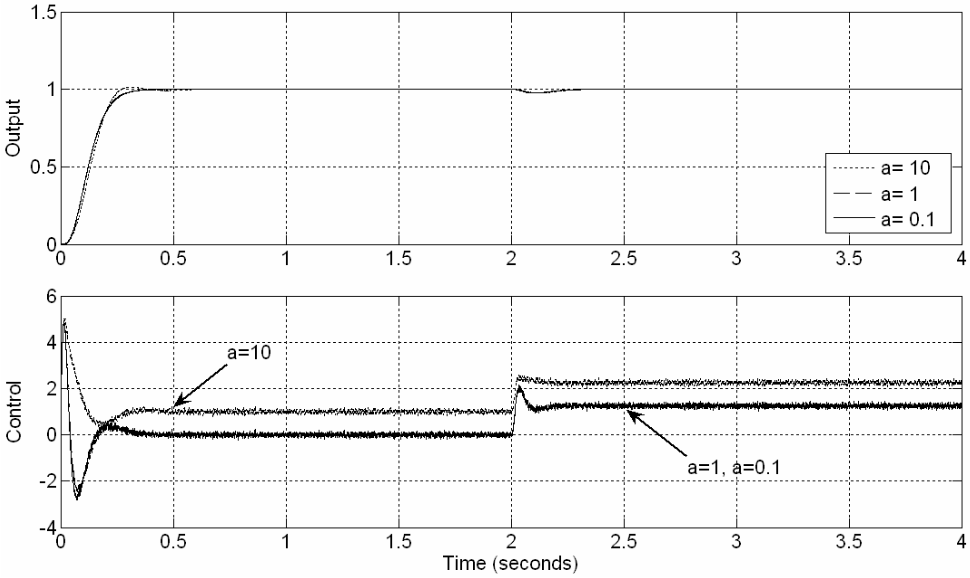


Figure 25: Three Lag System with a GPID Controller and Observer

4.2.6 Hardware Test

The industrial motion control test bed in Section 4.1.3 is used to verify the new control method, first in simulation and then in hardware. It was selected for its ability to accurately manipulate plant parameters and nonlinearities. A photo is shown in Figure 26 and a model is given in (4.40). The object is to rotate the larger load disk to a given angular position. A torque disturbance is applied by a separate servo motor at time t = 2 seconds and 0.1% white noise is injected into the output.

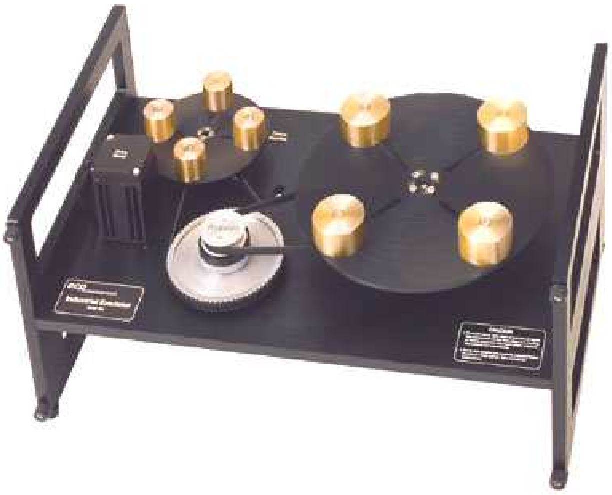


Figure 26: Industrial Motion Control Test Bed

Since the motor inductance is small enough with respect to its resistance that it is ignored, the transfer function of the system without a resonant mode and nonlinearities reduces to

$$\frac{x_l(s)}{V_c(s)} = \frac{b_m}{s(s+\omega_p)} \tag{4.80}$$

$$b_m = \frac{K_t R_p K_{pa}}{\left(R_a + K_{pa} K_{cf}\right) J_t} = 22.4, \quad \omega_p = \frac{K_t K_e}{\left(R_a + K_{pa} K_{cf}\right) J_t} = 1.12$$

A GPID controller and observer are selected to control the nonlinear model. The controller is digitized with a sample rate of 10 kHz using Euler integration.

$$\begin{aligned} u &= (\omega_c^3 \int (r-y)dt - 3\omega_c^2 z_1 - 3\omega_c z_2)/b \\ &\begin{cases} \dot{z}_1 = z_2 + 2\omega_o(y - z_1) \\ \dot{z}_2 = \omega_o^2(y - z_1) + bu \end{cases} \end{aligned} \tag{4.81}$$

The gain b = 25 was crudely estimated as the initial acceleration from a step response, as the case may be in practice instead of using the actual value b_m = 22.4.

A unique polynomial profile is used because it is a direct function of two common design specifications, the set point r_{sp} and the final time t_f. By defining $\tau = t / t_f$, the profile is given as

$$r(t) = \begin{cases} r_{sp}\left(6\tau^5 - 15\tau^4 + 10\tau^3\right), 0 \le \tau < 1 \\ r_{sp}, \; otherwise \end{cases}. \tag{4.82}$$

The profile also produces smooth position *p*, velocity *v*, and acceleration *a* trajectories while maintaining finite jerk *j*, as shown in Figure 27.

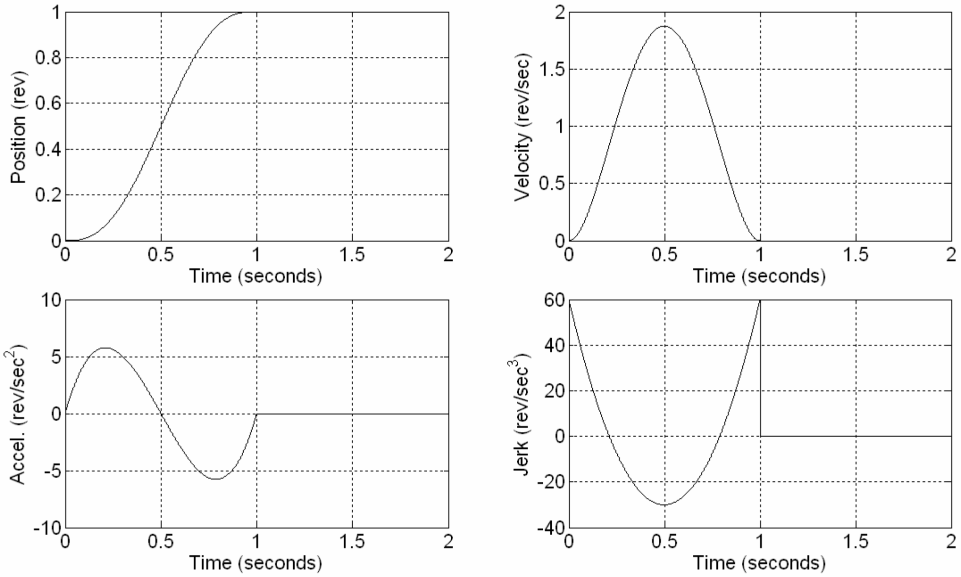


Figure 27: Polynomial Profile Trajectories

The maximum value of each trajectory is calculated by setting the derivative equal to zero and solving for the roots. They can be used to replace one of the other two design specifications, if required.

$$\begin{aligned} v_{\max} &= 15 r_{sp} / 8 t_f \\ a_{\max} &= 10\sqrt{3} r_{sp} / 3 t_f^2 \\ j_{\max} &= 60 r_{sp} / t_f^3 \end{aligned} \tag{4.83}$$

The design specifications call for one rotation of the load, or 2π radians, without exceeding a velocity of 20 radians per second. As a result, the profile is set for a settling time of 0.589 seconds.

Initially, $\omega_o = 10\omega_c$ and $\omega_c = 12.8$, which corresponds to the specified settling time of a third order step response. Then ω_c and ω_o are increased to 50 and 200, respectively, keeping the output smooth and the control signal noise level within ±100mV. The simulation results are shown below in Figure 28.

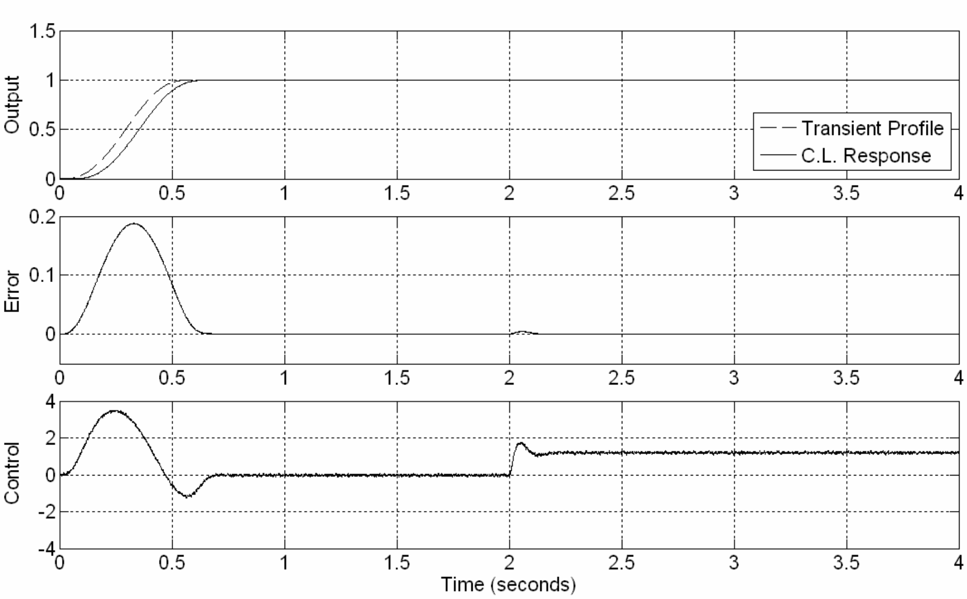


Figure 28: Simulation Test with a Nominal Load

The new controller and observer were then implemented in hardware. Due to a larger control signal and the absence of noise in the real system, the new challenge is to keep the control signal within ±3.5V. With this, ω_c and ω_o are increased to 50 and 100, respectively. The hardware results are shown in Figure 29.

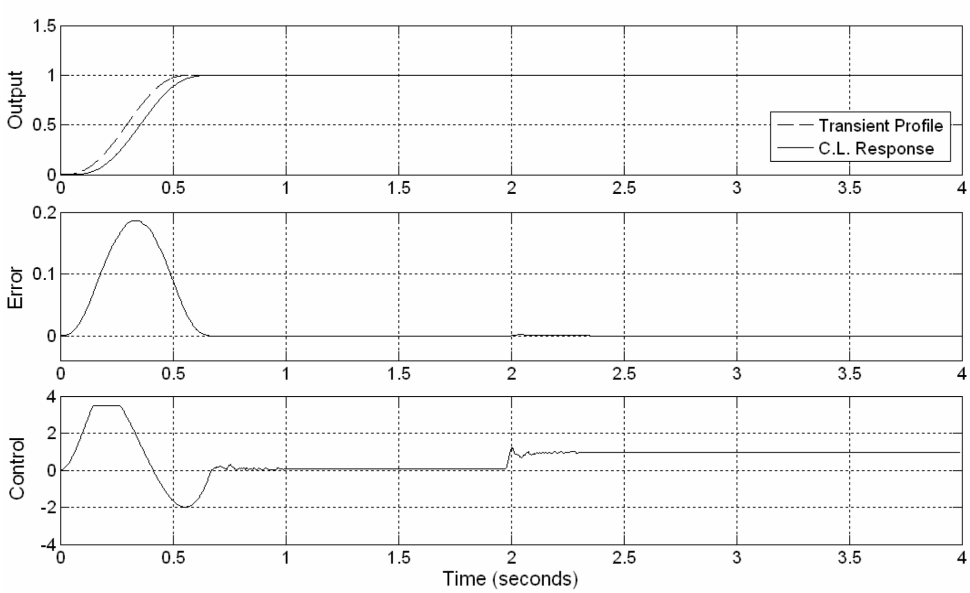


Figure 29: Hardware Test with a Nominal Load

In order to test for robustness, the reflected load inertia at the motor shaft was increased by a factor of nearly 8 by adding weights to the load. The results revealed no noticeable difference.

4.2.7 Combining ADRC and GPID

Note that the observer in (4.73) and (4.74) is merely an ESO$_{n,\,h}$ with $h = 0$ or zero extended states. The accuracy of the observer can be further increased when the effects of the disturbance are considered, meaning h>0. The discrete formulation is that of a CDESO. Disturbance rejection is increased when $\hat{f}$ is fed back to cancel f, as in ADRC.

$$u = u_0 - \hat{f} \tag{4.84}$$

Also note that the ADRC control law is a special case of (4.52) with $m = 0$ or zero extended states in the controller. Increasing the number of extended states in the controller and/or observer improves the disturbance rejection property of the controller. The result is a GPID controller with a CDESO where $\hat{f}$ can be fed back or not. However, it makes sense that for disturbance rejection to occur then $m+h>0$ and if m = 0, the ADRC case, then $\hat{f}$ should always be fed back. In other words, there has to be at least one extended state in the controller or one in the observer that is fed back.

4.3 Multivariable Control

Two methods of control are proposed for extension to multivariable; ADRC and GPID. Both are based on treating the plant as a set of decoupled SISO systems, each represented by $y_i^{(n)} = f_i + b_i u$ with disturbances that included the coupled dynamics. The difference is that ADRC is designed to estimate and cancel f_i while GPID is designed to reject f_i without explicitly estimating it. In this section, the proposed algorithms are extended to the general n^{th} order multivariable case with an arbitrary number of extended states.

4.3.1 Active Disturbance Rejection Control

Consider a system formed by a set of coupled n^{th} order input-output equations

$$\begin{aligned} y_1^{(n)} &= f_1 + b_1 U \\ &\vdots \\ y_q^{(n)} &= f_q + b_q U \end{aligned} \tag{4.85}$$

where $y_i^{(n)}$ denotes the n^{th} derivative of y_i. The input $U = [u_1, \cdots, u_p]^T$, the output $Y = [y_1, \cdots, y_q]^T$, and $b_i = [b_{i,1}, \cdots, b_{i,p}]$ for $i = 1,2,\cdots,q$ and $q \le p$. Each equation consists of two terms, the instantaneous b_iU and the dynamic $f_i(Y, \dot{Y}, \cdots, Y^{(n-1)}, t)$. All interactions between equations, internal dynamics, and external disturbances are considered part of $f_i(Y, \dot{Y}, \cdots, Y^{(n-1)}, t)$. The system is rewritten

$$Y^{(n)} = F + B_0 U \tag{4.86}$$

where $Y^{(n)} = [y_1^{(n)}, \cdots, y_q^{(n)}]^T$, $F = [f_1, \cdots, f_q]^T$, and $B_0 = [b_1^T, \cdots, b_q^T]^T$. Assuming that n is known and that B is an q×p approximation of B_0 where both are full row rank, a generalized disturbance is defined as $H \equiv F + (B_0 - B)U$. The system reduces to

$$Y^{(n)} = H + BU\,. \tag{4.87}$$

Extended State Observer

The idea is to estimate H and cancel it in real time, reducing the plant to a set of cascaded integrators. In order to represent the plant with a set of state equations, let $X = [X_1^T, X_2^T, \cdots, X_{n+h}^T]^T = [Y^T, \dot{Y}^T, \cdots, Y^{(n-1)T}, H^T, \cdots, H^{(h-1)T}]^T$ such that

$$\begin{cases} \dot{X}_1 = X_2 \\ \vdots \\ \dot{X}_{n-1} = X_n \\ \dot{X}_n = X_{n+1} + BU\,. \\ \dot{X}_{n+1} = X_{n+2} + \dot{H} \\ \vdots \\ \dot{X}_{n+h} = H^{(h)} \end{cases} \tag{4.88}$$

In state space form, the plant is represented by

$$\begin{aligned}\dot{X} &= \overline{A}X + \overline{B}U + \overline{E}H^{(h)} \\ \hat{Y} &= \overline{C}X\end{aligned} \tag{4.89}$$

where $\hat{X} = \left[\hat{X}_1^T, \hat{X}_2^T, \cdots, \hat{X}_{n+h}^T\right]^T$, 0_q and I_q are q×q zero and identity matrices, and $\overline{A}$ is an $q(n+h)$ dimensional square matrix.

$$\overline{A} = \begin{bmatrix} 0_q & I_q & 0_q & \cdots & 0_q \\ 0_q & 0_q & I_q & \cdots & 0_q \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_q & 0_q & 0_q & \cdots & I_q \\ 0_q & 0_q & 0_q & \cdots & 0_q \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} 0_{q\times p} \\ 0_{q\times p} \\ \vdots \\ B \\ 0_{qh\times p} \end{bmatrix}, \quad \overline{E} = \begin{bmatrix} 0_q \\ 0_q \\ \vdots \\ 0_q \\ I_q \end{bmatrix} \tag{4.90}$$
$$\overline{C} = \begin{bmatrix} I_q & 0_q & 0_q & \cdots & 0_q \end{bmatrix}$$

An observer is then designed from the state space model where $\overline{L} = [L_1, \cdots, L_{n+h}]^T$

$$\begin{aligned}\dot{\hat{X}} &= \overline{A}\hat{X} + \overline{B}U + \overline{L}(Y - \hat{Y}) \\ \hat{Y} &= \overline{C}\hat{X}\end{aligned} \tag{4.91}$$

That is, the state equations of the multivariable ESO become

$$\begin{cases} \dot{\hat{X}}_1 = \hat{X}_2 + L_1(Y - \hat{Y}) \\ \vdots \\ \dot{\hat{X}}_{n-1} = \hat{X}_n + L_{n-1}(Y - \hat{Y}) \\ \dot{\hat{X}}_n = \hat{X}_{n+1} + L_n(Y - \hat{Y}) + BU \\ \dot{\hat{X}}_{n+1} = \hat{X}_{n+2} + L_{n+1}(Y - \hat{Y}) \\ \vdots \\ \dot{\hat{X}}_{n+h} = L_{n+h}(Y - \hat{Y}) \end{cases} . \tag{4.92}$$

The observer gains $L_1, L_2, \cdots, L_{n+h}$ are q×q matrices, in general. However, for the sake of tuning simplicity, the gains are defined to form q parallel observer loops for $j = 1,2,\cdots,n+h$.

$$L_j = diag(l_{j,1}, l_{j,2}, \cdots, l_{j,q}) \tag{4.93}$$

Each loop then has its *n*+*h* poles placed in one location, for the sake of further simplification.

$$\lambda_o(s) = |sI - \overline{A} + \overline{L}\overline{C}| = \prod_{i=1}^{q}(s + \omega_{o,i})^{n+h} \tag{4.94}$$

Solving for each gain as a function of $\omega_{o,i}$ results in

$$l_{j,i} = \frac{(n+h)!}{j!(n+h-j)!}\omega_{o,i}^{j}\,. \tag{4.95}$$

<u>*ADRC Control Law*</u>

With B^+ defined as the right inverse of *B*, a disturbance rejection control law is applied to (4.87), effectively cancelling *H* at low frequencies.

$$U = B^+(U_0 - \hat{X}_{n+1}) \tag{4.96}$$

This allows a kind of feedback linearization and decoupling to occur which reduces the plant to a set of parallel *n*-integrator systems at low frequencies.

$$Y^{(n)} \approx U_0 \tag{4.97}$$

At this point, any number of control methods may be used. A simple GPID control law with no extended states is proposed

$$U_0 = K_0(Y^* - \hat{X}_1) - K_1\hat{X}_2 - \cdots - K_{n-1}\hat{X}_n \tag{4.98}$$

where Y^* is the desired trajectory for Y and the controller gains $K_0, K_1, \cdots, K_{n-1}$ are q×q matrices in general. However, for the sake of tuning simplicity, the controller gains are defined to form q parallel control loops for $j = 0,1,\cdots,n-1$.

$$K_j = diag(k_{j,1}, k_{j,2}, \cdots, k_{j,q}) \tag{4.99}$$

Each control loop has its n poles placed in one location, for the sake of further simplification.

$$\lambda_c(s) = | sI - \overline{A} + \overline{K} | = \prod_{i=1}^{q} (s + \omega_{c,i})^n \tag{4.100}$$

Solving for each gain as a function of $\omega_{c,i}$ results in

$$k_{j,i} = \frac{n!}{j!(n-j)!} \omega_{c,i}^{n-j}. \tag{4.101}$$

<u>Remarks</u>

1. Typically, a nonsingular B^{-1} can be approximated by a diagonal matrix of reciprocal elements, since inaccuracies in B can be accounted for in H.

2. The observer is simplified to remove B by substituting (4.96) into (4.92).

$$\begin{cases} \dot{\hat{X}}_1 &= \hat{X}_2 + L_1(Y - \hat{Y}) \\ \vdots & \\ \dot{\hat{X}}_{n-1} &= \hat{X}_n + L_{n-1}(Y - \hat{Y}) \\ \dot{\hat{X}}_n &= U_0 + L_n(Y - \hat{Y}) \\ \dot{\hat{X}}_{n+1} &= \hat{X}_{n+2} + L_{n+1}(Y - \hat{Y}) \\ \vdots & \\ \dot{\hat{X}}_{n+h} &= L_{n+h}(Y - \hat{Y}) \end{cases} \tag{4.102}$$

3. The commonly used SISO form of ADRC is, in fact, the q = 1 case.

Current Discrete ESO

To output the states, the ESO in (4.90) and (4.91) is rewritten as

$$\begin{aligned}\dot{\hat{X}} &= [\bar{A} - \bar{L}\bar{C}]\hat{X} + [\bar{B},\ \bar{L}][U,\ Y]^T \\ Y_c &= \hat{X}\end{aligned} \tag{4.103}$$

where Y_c is the state output. It can then be discretized, forming a multivariable CDESO

$$\begin{aligned}\hat{X}(k+1) &= [\bar{\Phi} - \bar{L}_p\bar{H}]\hat{X}(k) + [\bar{\Gamma}\ \ \bar{L}_p][U(k)\ Y(k)]^T \\ \bar{X}(k) &= [I_{q(n+h)} - \bar{L}_c\bar{H}]\hat{X}(k) + [0_{q(n+h)\times p}\ \ \bar{L}_c]\,[U(k)\ Y(k)]^T\end{aligned} \tag{4.104}$$

where $\bar{L}_c = [L_{c,1}, \cdots, L_{c,n+h}]^T$, $L_{c,j} = diag(lc_{j,1}, lc_{j,2}, \cdots, lc_{j,q})$, and similar notation is true for $\bar{L}_p$. The simplest way to determine the matrices is the case when (4.104) is equivalent to q parallel SISO loops. To show an example of how the matrices are directly extended from their SISO counterparts, the matrices for a CDESO$_{2,\ 1}$ as a result of ZOH become

$$\begin{aligned}\bar{\Phi} &= \begin{bmatrix} I_q & I_q T & I_q \frac{T^2}{2} \\ 0_q & I_q & I_q T \\ 0_q & 0_q & I_q \end{bmatrix},\ \bar{\Gamma} = \begin{bmatrix} BT \\ B \\ 0_{q\times p} \end{bmatrix},\ \begin{bmatrix} lc_{1,i} \\ lc_{2,i} \\ lc_{3,i} \end{bmatrix} = \begin{bmatrix} 1-\beta_i^3 \\ (1-\beta_i)^2(1+\beta_i)\frac{3}{2T} \\ (1-\beta_i)^3\frac{1}{T^2} \end{bmatrix}. \\ \bar{H} &= \begin{bmatrix} I_q & 0_q & 0_q \end{bmatrix}\end{aligned} \tag{4.105}$$

4.3.2 Generalized PID

Assuming n is known and B is a q×p approximation of some B_0 matrix where both are full row rank, consider the system represented by

$$Y^{(n)} = H + BU \tag{4.106}$$

where the input U is a p×1 matrix and the output Y is a q×1 matrix. All interactions between equations, internal dynamics, and external disturbances are considered part of H. Although it is clear that a multivariable ESO$_{n,\,h}$ from the previous section can be used with the GPID in a number of ways, the h = 0 case with no extended states is given to provide a simple example. Without knowledge of H, the design model is represented by a set of state equations

$$\begin{cases} \dot{X}_1 & = X_2 \\ \vdots & \\ \dot{X}_{n-1} & = X_n \\ \dot{X}_n & = BU \end{cases} \tag{4.107}$$

where $\left[X_1^T, X_2^T, \cdots, X_n^T\right]^T = \left[Y^T, \dot{Y}^T, \cdots, Y^{(n)T}\right]^T$. In state space form, the system is represented by

$$\begin{aligned} \dot{X} &= \overline{A}X + \overline{B}U \\ \hat{Y} &= \overline{C}X \end{aligned} \tag{4.108}$$

where $\hat{X} = \left[\hat{X}_1^T, \hat{X}_2^T, \cdots, \hat{X}_n^T\right]^T$, and $\overline{A}$ is a $q(n)$ dimensional square matrix.

$$\begin{aligned} \overline{A} &= \begin{bmatrix} 0_q & I_q & 0_q & \cdots & 0_q \\ 0_q & 0_q & I_q & \cdots & 0_q \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_q & 0_q & 0_q & \cdots & I_q \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} 0_{q\times p} \\ 0_{q\times p} \\ \vdots \\ B \end{bmatrix}, \quad \overline{L} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \\ \overline{C} &= \begin{bmatrix} I_q & 0_q & 0_q & \cdots & 0_q \end{bmatrix} \end{aligned} \tag{4.109}$$

An observer is then designed from the state space model.

$$\begin{aligned}\dot{\hat{X}} &= \overline{A}\hat{X} + \overline{B}U + \overline{L}(Y - \hat{Y}) \\ \hat{Y} &= \overline{C}\hat{X}\end{aligned} \tag{4.110}$$

That is, the state equations of the multivariable observer become

$$\begin{cases}\dot{\hat{X}}_1 &= \hat{X}_2 + L_1(Y - \hat{Y}) \\ \vdots & \\ \dot{\hat{X}}_{n-1} &= \hat{X}_n + L_{n-1}(Y - \hat{Y}) \\ \dot{\hat{X}}_n &= BU + L_n(Y - \hat{Y})\end{cases} \tag{4.111}$$

The observer gains $L_1, L_2, \cdots, L_n$ are q×q matrices, in general. However, for the sake of tuning simplicity, the gains are defined to form q parallel observer loops for $j = 1,2,\cdots,n$.

$$L_j = diag(l_{j,\,1}, l_{j,\,2}, \cdots, l_{j,\,q}) \tag{4.112}$$

Each loop then has its n poles placed in one location, for the sake of further simplification.

$$\lambda_o(s) = \mid sI - \overline{A} + \overline{L}\overline{C} \mid = \prod_{i=1}^{q}(s + \omega_{o,\,i})^n \tag{4.113}$$

Solving for each gain as a function of $\omega_{o,\ i}$ results in

$$l_{j,\,i} = \frac{n!}{j!(n-j)!}\omega_{o,\,i}^{j}\,. \tag{4.114}$$

<u>GPID Control Law</u>

With B^+ defined as the right inverse of B, and Y^* is a q×1 reference input, the following GPID control law is proposed.

$$U = B^{+}\begin{pmatrix} K_{-m}\overset{(m)}{\int}(Y^{*}-Y)-K_{1-m}\overset{(m-1)}{\int}Y-\cdots-K_{-1}\int Y \\ -K_{0}\hat{X}_{1}-\cdots-K_{n-1}\hat{X}_{n} \end{pmatrix}. \tag{4.115}$$

The gains are defined to form q parallel control loops for $j=-m,1-m,\cdots,n-1$.

$$K_{j} = diag(k_{j,\,1},k_{j,\,2},\cdots,k_{j,\,q}) \tag{4.116}$$

Each control loop has its n+m poles placed in one location, for the sake of further simplification.

$$\lambda_{c}(s) = \mid sI-\overline{A}+\overline{K}\mid = \prod_{i=1}^{q}(s+\omega_{c,\,i})^{n+m} \tag{4.117}$$

Solving for each gain as a function of $\omega_{c,\ i}$ results in

$$k_{j,\,i} = \frac{(n+m)!}{(n-j)!(m-j)!}\omega_{c,\ i}^{n-j}. \tag{4.118}$$

4.3.3 Gain and Order Estimation

Various control methods have been proposed here to control a system of the form $Y^{(n)} = F + BU$ where the relative order n is known and the high frequency gain B is within 50% of its true value. The next step is to create an algorithm that quickly estimates these parameters by extracting them from plant input-output data prior to control. In this section, a method is proposed for SISO systems that are initially at rest, followed by a discussion on how the method is applied to MIMO systems.

Gain Estimation

For clarity, consider the system in (4.21) represented as a function of time.

$$y^{(n)}(t) = f(t) + bu(t) \tag{4.119}$$

Assuming the system is at rest and there are no external disturbances at t = 0, then $f(0) = 0$ and all derivatives of y are zero. Directly after the system is excited through $u(t)$ at time $t = \varepsilon$, where ε is a very small number, then $f(\varepsilon) \approx 0$ and

$$y^{(n)}(\varepsilon) \approx bu(\varepsilon)\,. \tag{4.120}$$

With minor manipulation [73], it is clear that

$$b \approx y^{(n)}(\varepsilon)/u(\varepsilon) \tag{4.121}$$

and this approximation in the frequency domain becomes

$$b \approx s^{(n)} y(s)/u(s), \quad \forall s\,. \tag{4.122}$$

The problem with this method is in the techniques used to approximate $y^{(n)}(\varepsilon)$. Such techniques are either noisy and require a step input, as is the case when using

$$\dot{y}(k) \approx \frac{y(k-1) - y(k)}{T}, \tag{4.123}$$

or they are inaccurate as is the case when using linear filters to approximate derivatives at $t = \varepsilon$ due to the exponential characteristic of the filter's response in the time domain.

$$y^{(n)}(s) \approx s^{(n)} H(s) y(s), \quad H(s) = 1/(\tau s + 1)^{n+1} \tag{4.124}$$

Since noise filtering improves as the time constant τ is reduced, the idea is to also filter $u(s)$ with $H(s)$ before dividing the two in the time domain so that the exponential characteristic of the filter is canceled. The new method becomes

$$b \approx \frac{L^{-1}\left\{s^{(n)}H(s)y(s)\right\}}{L^{-1}\left\{H(s)u(s)\right\}} \tag{4.125}$$

where the inverse Laplace Transformation is denoted as $L^{-1}\{\cdot\}$. This way the time constant of $H(s)$ can be adjusted to filter noise without affecting the estimation of b. Experimental tests have shown that an input other than a step works well if equipment damage becomes an issue, however, a step will produce the best estimation because it excites the high frequency characteristics of the system before the slower dynamics in f respond, which would make the approximation in (4.120) less accurate.

When the filters are digitized using a bilinear transformation,

$$s \approx (2z-2)/(Tz+T) \tag{4.126}$$

each integrator in $H(s)$ has a one time step propagation delay through the filter when it is realized in Direct Transposed II form [74], [75]. As a result, the estimation of b should be valid after time step $k = n+1$.

Order Estimation

The determination of order involves a bit of discrete timing. First, the previous method is applied over $k = n+2$ time steps to determine b assuming $n = 1$. In the next time step, the disturbance is back-calculated over $k = 1,2,\cdots,n+2$ time steps using b as

$$f_n(k) = y^{(n)}(k) - b_n u(k)\,. \tag{4.127}$$

The process is repeated assuming $n = 1,2,\cdots,r$ which generates corresponding (f_n, b_n) pairs for each $n = 1,2,\cdots,r$ where f_n is a discrete signal vector and b_n is a scalar. The integral absolute error of each disturbance is then calculated as

$$g_n = \sum_{k=1}^{n+2} \left| f_n(k) \right| \tag{4.128}$$

The idea is to determine the n producing the smallest disturbance because it will require the least amount of control effort to overcome. For the purpose of control, the effective relative order of the system n^* is determined by finding the minimum g_n.

$$n^* : g_{n^*} = \min(g_1, g_2, \cdots, g_5) \,. \tag{4.129}$$

The corresponding b_{n^*} is the gain used in the control system. Even with resonances, most systems should have dominant dynamics less than 5th order. As a result r = 5 is sufficient and the algorithm can compute b and n in r + 3 time steps.

Furthermore, time delay of a system can be measured by counting the number of time steps after the excitation of $u(k)$ that the output $y(k)$ becomes nonzero. This should be taken into account for the algorithm to be accurate, meaning the time step count shouldn't start until the output becomes nonzero.

Extension to MIMO Systems

For MIMO systems, the gain estimation method can be applied to each input-output pair, forming a gain matrix. The matrix is determined by assuming the order is known for each case of $n = 1, 2, \cdots, r$. Next, F is back-calculated for each case. The n resulting in the smallest $\|F\|$ becomes the n that is used to control the system, along with its corresponding B.

Outlined in Section 2.3.2, the standard perturbation method built into MAPSS can be used to find the B matrix, although some problems need to be addressed. First, the plant must be first order. The CLM runs at a sample rate of 2500 Hz, but the perturbation method runs at the much lower sample rate of the controller, 50 Hz. The problem arises when fast dynamics are sampled too slowly and they appear to be instantaneous or part of the *D* matrix. As a fix to the problem, the perturbation method may need to run at 2500 Hz. The biggest problem, however, is that this method can't be used to determine the high frequency gain *B* because a nonzero *F* is unaccounted for in the calculation, i.e. the proposed methods can't be used with systems that are not at rest, such as the jet engine.

CHAPTER V

JET ENGINE APPLICATION

Several new disturbance rejection techniques have been proposed. In this chapter, a generic design procedure is given, using the application of multivariable ADRC to the MAPPS jet engine as an example. Test conditions are then discussed for the jet engine, followed by simulation results that compare the new algorithm to the current one, showing that similar performance can be achieved with much less design effort.

5.1 Design Procedure

The design procedure for applying any of the new disturbance rejection techniques is uniquely characterized by the plant representation.

$$Y^{(n)} = H + BU \tag{5.1}$$

In (5.1), the size of the input vector *U* and output vector *Y* should be known. The design procedure involves the determination of *n* and *B* as well as a method of tuning the controller. A generic procedure is given, followed by a detailed explanation of each step with examples specific to the jet engine application.

Procedure Overview

1. Determine the number of inputs *p* and outputs *q* of the system. Use multi-mode control if $q > p$.
2. Determine the high frequency gain *B* of the system.
3. Determine the relative order *n* of the system. If *n* is unknown, begin by assuming $n = 1$.
4. Determine the number of extended states *h*. A value of $h = 1$ is usually sufficient.
5. Apply the new algorithm to the system.
6. Run a closed loop simulation or hardware test in order to tune the controller and observer bandwidths.
7. Run a closed loop simulation or hardware test in order to tune B^+, the right inverse of *B*. Repeat step six as required.

Step One

The first step is to determine the number of control inputs *p* and controlled outputs *q* of the plant. If $q > p$, then multi-mode control should be used to make $q_j \leq p_j$ for each of the *j* sub-controllers. It is preferable to make $q_j = p_j$ which produces a square *B* matrix that allows the diagonal elements to become tuning parameters. Note that when *B* is a square matrix, $B^+ = B^{-1}$. A diagonal *B* matrix also permits the new technique to be reduced to multiple SISO techniques.

The jet engine in MAPPS, for example, has 3 actuator inputs that control 7 performance parameter outputs. As a result, the jet engine controller consists of four separate regulators, each controlling only 3 outputs at a time. In this research, a simple form of multivariable ADRC using Euler integration is applied to the 3-input 3-output low speed regulator section and tested in simulation. This approach will isolate the affects that blending of multiple modes may have on the results.

Step Two

The second step is to determine the high frequency gain *B* of the system. This matrix will drastically change for different values of relative order *n*. The trick is that *n* must be known in order to determine *B*, and *B* must be known to determine *n*, a circular argument. As a result, steps two and three are interchangeable and an iterative process may be used in finding *n* and *B*. Nevertheless, there is only a problem if both are unknown. If this is the case, use the identity matrix for *B* to first determine *n* and then iterate. The *B* matrix can also be tuned in step seven. In practice, B^+ needs to be initially within fifty percent of its true value and such a broad range is frequently known. However, if it is unknown or the system is too complex, then various system identification techniques or the technique in Section 4.3.3 can be used.

In MAPSS, the control signals are scaled to produce the proper units for each actuator input allowing each control signal to be within the same relative range. Thus, a logical starting point for *B* in the low speed regulator is the identity matrix.

Step Three

The third step is to determine the relative order *n* of the plant. The overall structure of the observer and controller depends on *n* which may or may not be the actual order of the system, depending on which dynamics are dominant. The idea is to find *n* = 1, 2, or 3 that produces the smallest *H*, the smallest control signal, or the best closed loop results. Sometimes *n* is known or can be derived from a model of physical relationships. However, if it is not known and can not be derived because the system is unknown or too complex, then the technique in Section 4.3.3 can be used as long as the plant can be excited from rest. If none of the above techniques work, the last resort is trial and error. In this situation, start by assuming the system is first order and complete the remaining steps. Then assume the system is second order and repeat the process to see if the results are better. Then try third order, etc. Lower order is usually better.

Another consideration is that *n* can be determined for each plant input-output pair. For a particular output, the input yielding the lowest order with the highest gain is the most direct form of control and therefore should be used.

Since not all of the engine's states are measurable in MAPPS, the model for low speed regulation is represented as a nonlinear input-output vector function. Without explicit knowledge of system order, the simplest and lowest order case is first attempted.

$$\dot{Y}_1 = F(Y_1, U, t) \tag{5.2}$$

When a 3×3 matrix *B* is used to approximate the actual high frequency gain B_0, the signal *H* is defined as

$$H \equiv F(Y, U, t) - B_0 U \,. \tag{5.3}$$

The system then reduces to a form that has distinct terms to represent any internal or external dynamics and an instantaneous input.

$$\dot{Y} \approx H + BU \tag{5.4}$$

After running simulation tests at higher orders, first order was found to be sufficient for MAPPS. This also makes sense since the CLM is represented as a first order state space equation and the actuator dynamics are fast enough to be neglected.

Step Four

The fourth step is to determine the number of extended states *h*. This affects the overall structure of the ESO. For ADRC, select *h* = 1, 2, or 3 based on the system type of the disturbance *H* or an external disturbance. The assumption *h* = 1 will suffice in most cases and therefore is used in the remaining examples for the sake of clarity. Similar is true for determining the number of extended states *m* in the control law when using generalized PID.

Step Five

The fifth step is to apply the new algorithm to the system. The overall configuration is shown in Figure 30.

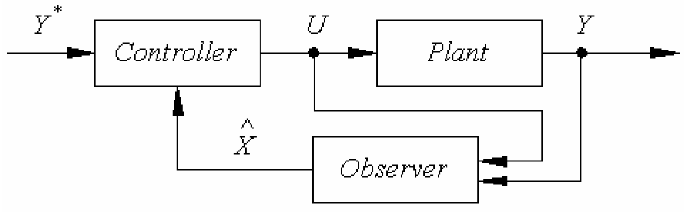


Figure 30: Control Configuration

The structure of the observer and controller depends on the integer values selected for n and h. The most frequently used cases are now explicitly given for multivariable ADRC with Euler integration. In general, U is a p×1 vector, Y is a q×1 vector, B is a q×p matrix, and L_j and K_j are q×q matrices.

When $n = 1$ and $h = 1$, the ESO equations become

$$\begin{aligned}\dot{\hat{X}}_1 &= \hat{X}_2 + L_1(Y - \hat{X}_1) + BU \\ \dot{\hat{X}}_2 &= L_2(Y - \hat{X}_1)\end{aligned} \tag{5.5}$$

and the controller is represented by

$$U \; = B^{+}(K_p(Y^{*} - \hat{X}_1) - \hat{X}_2)\,. \tag{5.6}$$

By applying (5.5) and (5.6), the control configuration is then shown in Figure 31.

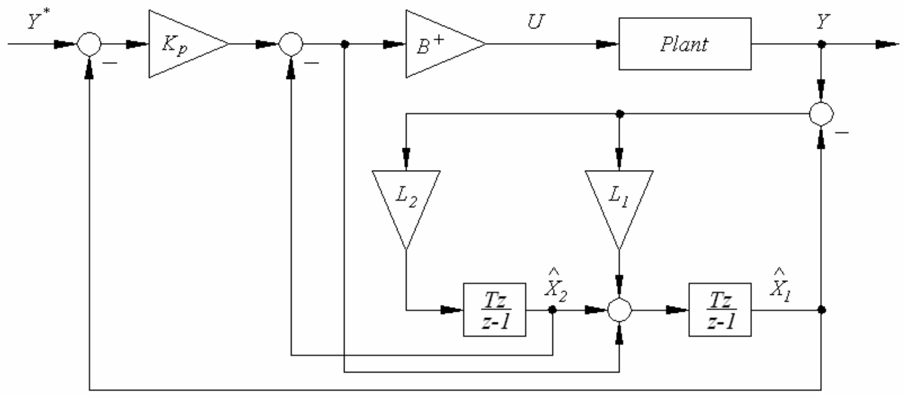


Figure 31: ADRC for a First Order System

Notice that since the input to B^+ is essentially BU, it is used as an input to the ESO instead of multiplying U by B. In doing so, there is only one matrix to adjust containing elements of B and it acts to scale the plant and allow the rest of the algorithm to be designed for a unity gain plant.

When $n = 2$ and $h = 1$, the ESO equations become

$$\begin{aligned} \dot{\hat{X}}_1 &= \hat{X}_2 + L_1(Y - \hat{X}_1) \\ \dot{\hat{X}}_2 &= \hat{X}_3 + L_2(Y - \hat{X}_1) + BU \\ \dot{\hat{X}}_3 &= L_3(Y - \hat{X}_1) \end{aligned} \tag{5.7}$$

and the controller is represented by

$$U \;= B^+(K_p(Y^* - \hat{X}_1) - K_d\hat{X}_2 - \hat{X}_3). \tag{5.8}$$

By applying (5.7) and (5.8), the control configuration is then shown in Figure 32.

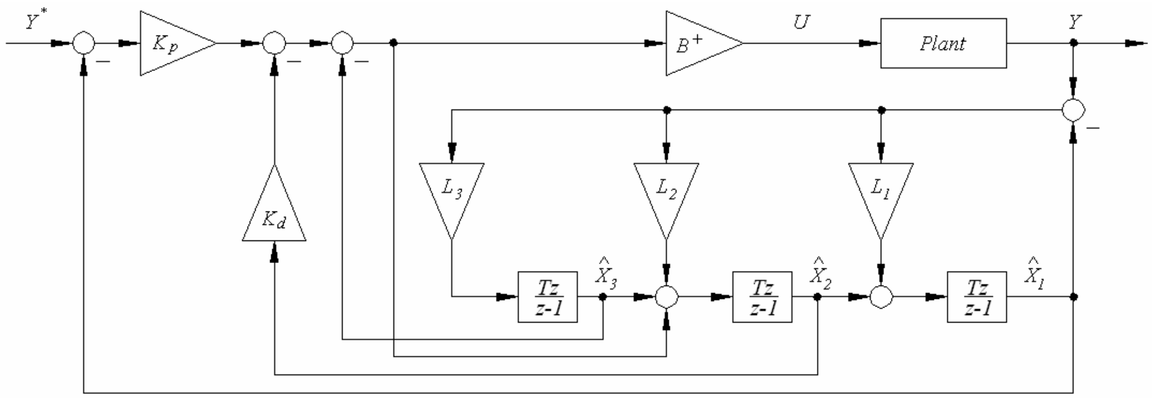


Figure 32: ADRC for a Second Order System

The first order plant in (5.4) used for low speed regulation is represented by state equations where the extended state X_2 is assigned to track the general disturbance H

$$\begin{aligned} \dot{X}_1 &= X_2 + BU \\ \dot{X}_2 &= \dot{H} \end{aligned} \tag{5.9}$$

and the 3×1 state vectors are defined as $X = [X_1^T, X_2^T]^T = [Y^T, H^T]^T$. By also defining the estimated states as $\hat{X} = [\hat{X}_1^T, \hat{X}_2^T]^T$, an ESO is designed from (5.9).

$$\begin{aligned}\dot{\hat{X}}_1 &= \hat{X}_2 + L_1(Y - \hat{X}_1) + BU \\ \dot{\hat{X}}_2 &= L_2(Y - \hat{X}_1)\end{aligned} \tag{5.10}$$

A disturbance rejection control law is then implemented

$$U = B^{+}(U_0 - \hat{X}_2) \tag{5.11}$$

to decouple the plant, reducing it to three parallel integrators at low frequencies

$$\dot{Y} \approx U_0 \tag{5.12}$$

whereby a simple proportional control law is applied.

$$U_0 = K_1(Y^{*} - \hat{X}_1) \tag{5.13}$$

The entire algorithm consists of (5.5) and (5.6) and is shown implemented in Figure 31.

$$\begin{aligned}\dot{\hat{X}}_1 &= \hat{X}_2 + BU + L_1(Y - \hat{X}_1) \\ \dot{\hat{X}}_2 &= L_2(Y - \hat{X}_1) \\ U \ &= B^{+}(K_1(Y^{*} - \hat{X}_1) - \hat{X}_2)\end{aligned} \tag{5.14}$$

The original jet engine regulators incorporate PID controllers that are subject to integrator windup because the integrator input is a function of the controller error $R - Y$ where convergence to zero is affected by plant saturation. An interesting benefit of replacing these regulators with ADRC is that the integrators within ADRC do not wind up because their inputs are a function of the observer error $Y - \hat{X}_1$ where convergence to zero is not affected by plant saturation. As a result, additional anti-windup mechanisms are no longer required.

Step Six

The sixth step is to run the closed loop system in order to tune the controller and observer bandwidths. In general, L_j and K_j are q×q matrices. However, when L_j and K_j are selected as diagonal matrices, the ADRC algorithm reduces to a set of SISO controllers, one to control each output. An example is given in Figure 33 for a three-output system. A tuning procedure for this type of configuration is proposed.

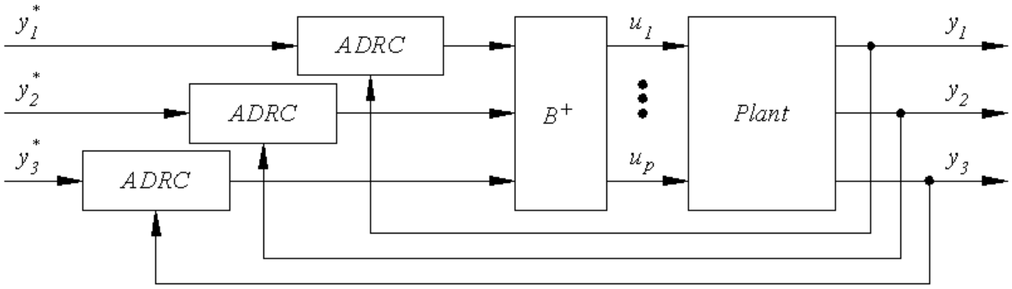


Figure 33: Unity Gain SISO Loops

When n = 1 and h = 1, the resulting observer and controller gain matrices become

$$\begin{aligned} L_1 &= diag(2\omega_{o,1}, 2\omega_{o,2}, \cdots, 2\omega_{o,q}) \\ L_2 &= diag(\omega_{o,1}^2, \omega_{o,2}^2, \cdots, \omega_{o,q}^2) \end{aligned} \tag{5.15}$$

$$K_p = diag(\omega_{c,1}, \omega_{c,2}, \cdots, \omega_{c,q}). \tag{5.16}$$

When n = 2 and h = 1, the resulting observer and controller gain matrices become

$$\begin{aligned} L_1 &= diag(3\omega_{o,1}, 3\omega_{o,2}, \cdots, 3\omega_{o,q}) \\ L_2 &= diag(3\omega_{o,1}^2, 3\omega_{o,2}^2, \cdots, 3\omega_{o,q}^2) \\ L_3 &= diag(\omega_{o,1}^3, \omega_{o,2}^3, \cdots, \omega_{o,q}^3) \end{aligned} \tag{5.17}$$

$$\begin{aligned} K_p &= diag(\omega_{c,1}^2, \omega_{c,2}^2, \cdots, \omega_{c,q}^2) \\ K_d &= diag(2\omega_{c,1}, 2\omega_{c,2}, \cdots, 2\omega_{c,q}) \end{aligned}. \tag{5.18}$$

At this point, a procedure to tune $\omega_{c,i}$ and $\omega_{o,i}$ for each output i is proposed as follows, dropping the i subscript for the sake of clarity.

1. Set ω_c using (4.59) according to initial specifications.

2. Set $\omega_o=2\sim10\omega_c$ as a rule of thumb. The exact relation will be based on the proximity of the desired closed loop bandwidth to the dominant poles of the system, resonant frequencies, and noise.

3. Run the closed loop system and increase ω_c and ω_o together to a point just before oscillation appears in the control signal.

4. Adjust the relation between ω_c and ω_o to meet the design specifications for noise level and disturbance rejection.

In MAPPS, all three observer bandwidths are set equal, $\omega_{o,1}=\omega_{o,2}=\omega_{o,3}=\omega_o$, for the sake of simplicity and proof of concept. This makes the observer gain matrices a function of one parameter.

$$L_1 = 2\omega_o I_3, \quad L_2 = \omega_o^2 I_3 \tag{5.19}$$

All three controller bandwidths are also set equal, $\omega_{c,1}=\omega_{c,2}=\omega_{c,3}=\omega_c$, for the same reason, making the controller gain matrix a function of one parameter as well.

$$K_p = \omega_c I_3 \tag{5.20}$$

<u>Step Seven</u>

The seventh step is to run the closed loop system in order to tune B^+. Although B is determined using system identification techniques in step two, it acts as a control signal gain that directly affects overshoot of the closed loop system. As a result, *B* can also be tuned by adjusting its elements to the point just before overshoot appears in the output. It is preferable to have as many inputs as outputs to facilitate tuning. This is demonstrated by first expanding the vector products $U = [u_1, \cdots, u_p]^T$ and $b_i = [b_{i,1}, \cdots, b_{i,p}]$ in (4.85).

$$\begin{aligned} y_1^{(n)} &= f_1 + b_{1,1}\,u_1 + \cdots + b_{1,q}\,u_q \\ y_2^{(n)} &= f_2 + b_{2,1}\,u_1 + \cdots + b_{2,q}\,u_q \\ &\vdots \\ y_q^{(n)} &= f_q + b_{q,1}\,u_1 + \cdots + b_{q,q}\,u_q \end{aligned} \tag{5.21}$$

In the i^{th} state equation, the input u_i is used to control the output y_i, and the rest of the inputs are combined into a new disturbance h_i.

$$\begin{aligned} h_1 &= f_1 + b_{1,2}\,u_2 + \cdots + b_{1,q}\,u_q \\ h_1 &= f_2 + b_{2,1}\,u_2 + b_{2,3}\,u_2 + \cdots + b_{2,q}\,u_q \\ &\vdots \\ h_q &= f_q + b_{q,1}\,u_1 + \cdots + b_{q-1,q-1}\,u_{q-1} \end{aligned} \tag{5.22}$$

The plant is rewritten where only the diagonal elements of *B* are considered.

$$\begin{aligned} y_1^{(n)} &= h_1 + b_{1,1}\,u_1 \\ y_2^{(n)} &= h_2 + b_{2,2}\,u_2 \\ &\vdots \\ y_q^{(n)} &= h_q + b_{q,q}\,u_q \end{aligned} \tag{5.23}$$

The inverse of each diagonal element then becomes a tuning parameter of an individual SISO control loop.

$$B^{-1} = diag(b_1^{-1}, b_2^{-1}, \cdots, b_q^{-1}) \tag{5.24}$$

An example is given in Figure 34 for a three-output system.

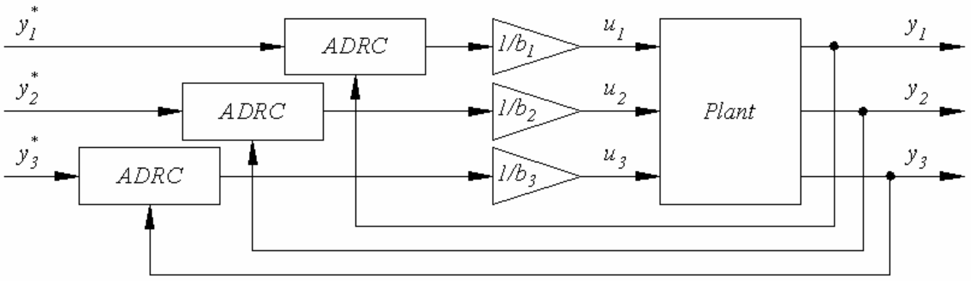


Figure 34: Multiple SISO Loops

Since each jet engine regulator has 3 inputs and 3 outputs and the system parameters are unknown, the plant used for low speed regulation was represented by (5.23) and the diagonal elements of B^{-1} were tuned. The identity matrix was selected as a starting point. The relative signs of each diagonal element were next determined, followed by magnitude adjustment.

5.2 Test Conditions

Turbofan engine performance varies from engine to engine due to manufacturing tolerances and deterioration from extended use [76]. Even though degradation may eventually require an engine to be overhauled as limits are reached, the engine control system should be robust enough to keep the engine operating within safe limits for several thousand flight cycles. With repeated use, the engine components wear and performance is degraded. For example, turbine blades erode and clearances open up. In order to achieve the same level of thrust as a new engine, a deteriorated engine must run hotter and/or faster [77]. This shift from nominal operation increases with use, and eventually reaches the point where performance can not be maintained without compromising the safety of the engine or the life of its components [77]. The degradation in performance is simulated in MAPSS by adjusting ten health parameters.

In most turbofan engines, thrust is calculated as a function of regulated and non-regulated variables, since it cannot be directly measured [78]. Although regulated variables are maintained at their set points regardless of engine degradation, non-regulated parameters shift from their nominal values with deterioration [2]. As a result, the closed-loop performance of the current model-based controller suffers as the engine wears. One of the objectives here is to control the transient thrust response of a deteriorated engine, making it behave as close to a new engine as possible.

Gas path analysis is a diagnostic technique that is used to estimate and trend health parameters by examining shifts in component health based upon gas path sensor measurements, i.e. pressures, temperatures, rotor speeds, and the known aero-thermodynamic relationships that exist between them [79]. As in [77], the health parameters follow an average degradation profile over the life span of the engine

$$p_i = a_i(1 - \exp(-b_i t_{eff})) + c_i t_{eff} \tag{5.25}$$

where a_i, b_i, and c_i are constants for each health parameter and t_{eff} represents the physical age of the engine in effective flight cycles rather than it chronological age. The initial exponential rise is intended to simulate rub-in and new engine deterioration mechanisms [76]. As the engine ages, the health parameter degradation tends to become more linear, as shown in references [80] and [81].

The component degradation values in percent resulting from health parameter changes are shown in the following table. They reflect moderate to beyond severe degradation [76] such as what might occur when the engine is due for an overhaul or when the engine is placed in a harsh desert environment [77].

TABLE VIII: DEGRADATION VALUES DUE TO HEALTH PARAMETER CHANGES

Flight Cycles	**Fan**		**LPC**		**HPC**		**HPT**		**LPT**	
t_{eff}	*Flow %*	*%*	*Flow %*	*%*	*Flow %*	*%*	*Flow %*	*%*	*Flow %*	*%*
0	0	0	0	0	0	0	0	0	0	0
3000	-2.04	-1.50	-2.08	-1.46	-3.91	-2.94	+1.76	-2.63	+0.26	-0.538
3750	-2.443	-1.838	-2.56	-1.748	-6.448	-4.555	-1.96	-2.925	+0.3	-0.673
4500	-2.845	-2.175	-3.04	-2.035	-8.985	-6.17	+2.16	-3.22	+0.34	-0.808
5250	-3.248	-2.512	-3.52	-2.323	-11.52	-7.785	+2.37	-3.515	+0.38	-0.943
6000	-3.65	-2.85	-4	-2.61	-14.06	-9.4	+2.57	-3.81	+0.42	-1.078

The level of degradation is characterized by effective cycles t_{eff} where zero cycles is a new engine with no degradation, 3000 cycles is moderate degradation, 4500 cycles is heavy degradation, and 5250 cycles is severe degradation.

Test operating points were selected to cover a large portion of the entire MAPSS flight envelope and most of the subsonic range. Test point #1 represents ground idle conditions where the *pla* is stepped from 21 to 35 for takeoff. The rest of the test points, with the exception of #6, represent the majority of subsonic power conditions. Test point #6 approaches high speed cruise conditions nearing the limits of the low speed controller.

TABLE IX: TEST OPERATING POINTS WITHIN THE MAPSS ENVELOPE

Op Pt.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
alt	0K	20K	20K	36,089	36,089	36,089	20K	40K	40K	40K
xm	0	0.5	0.8	0.5	0.8	1.0	0.3	0.3	0.5	0.8
pla	21-35	30-35	30-35	30-35	30-35	32-37	30-35	30-35	30-35	30-35

5.3 Simulation Results

The redesigned low speed regulator in (5.14) was digitized using Euler integration. MAPSS is a multi-rate simulation package where the engine sample time is fixed at 0.0004 seconds and the controller sample time is fixed at 0.02 seconds. For proof of concept, the new ADRC controller and the supplied nominal controller were simulated at each of the first three operating points in Table IX. The results from the nominal controller are used as a reference to compare the performance of the ADRC controller with. The goal here was not to show that one controller is better in performance over the other but merely that they are comparable in performance and ADRC is very simple to design, especially since the exact method of tuning the nominal controller is not known. The new ADRC controller was then simulated at the next three operating points in Table IX to show how it is able operate over a substantial range of the low speed regulator. The last four operating points were omitted because they show more of the same types of responses, i.e. no new information.

All simulations were conducted for each of the six levels of degradation in Table VIII. As shown in Table X, they are labeled as Run 1 through Run 6 in each simulation.

TABLE X: DEGRADATION TEST RUNS

Run#	1	2	3	4	5	6
t_{eff}	0	3000	3750	4500	5250	6000

Although the high speed regulator and the other two fan safety regulators were not tested, similar performance is expected. The results are shown in Figures 35 through 52.

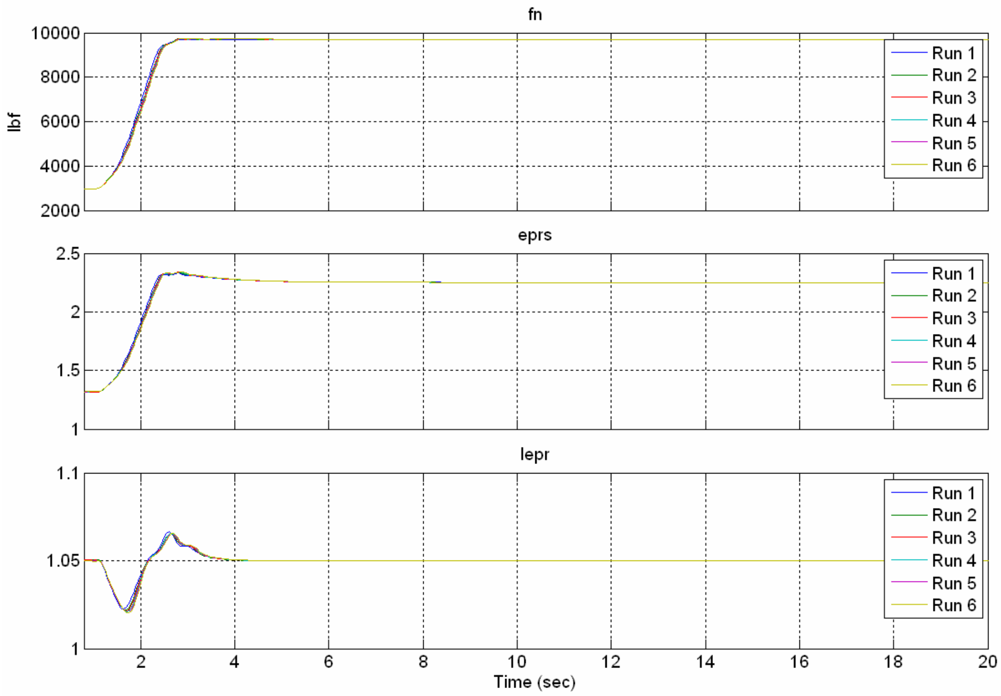


Figure 35: ADRC Controlled Variables at Operating Point #1

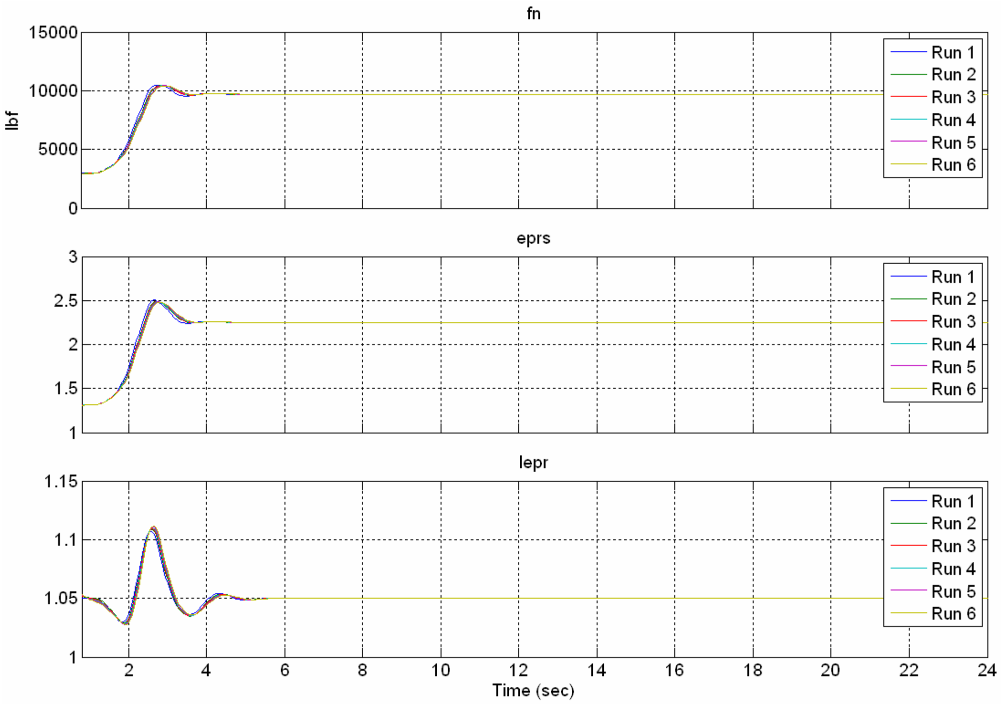


Figure 36: Nominal Controlled Variables at Operating Point #1

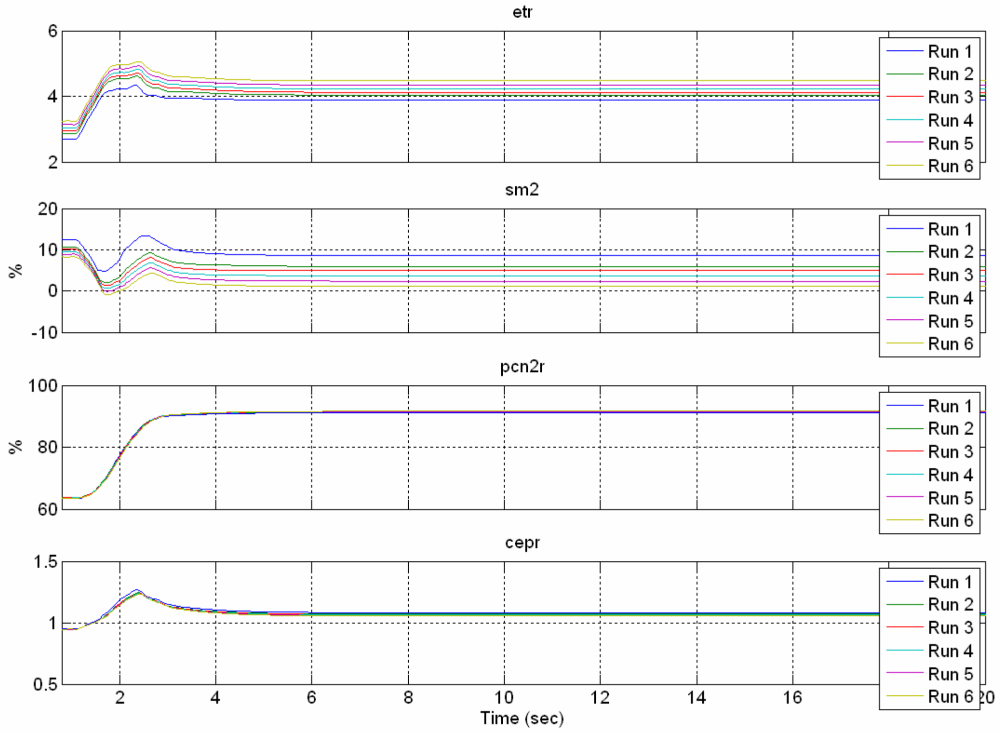


Figure 37: ADRC Uncontrolled Variables at Operating Point #1

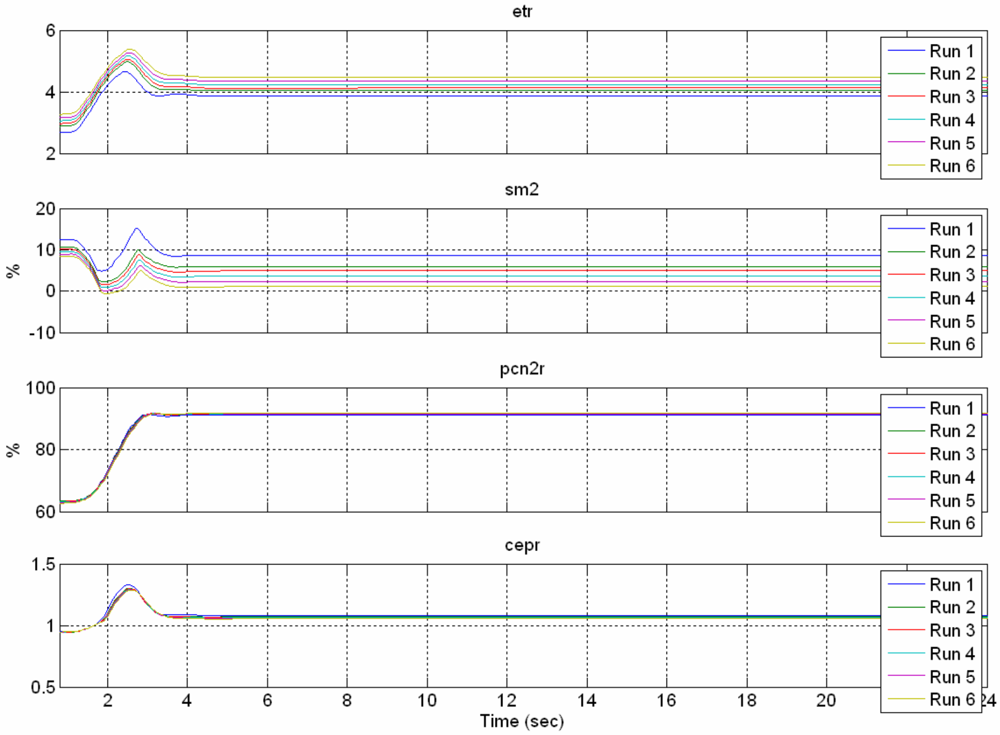


Figure 38: Nominal Uncontrolled Variables at Operating Point #1

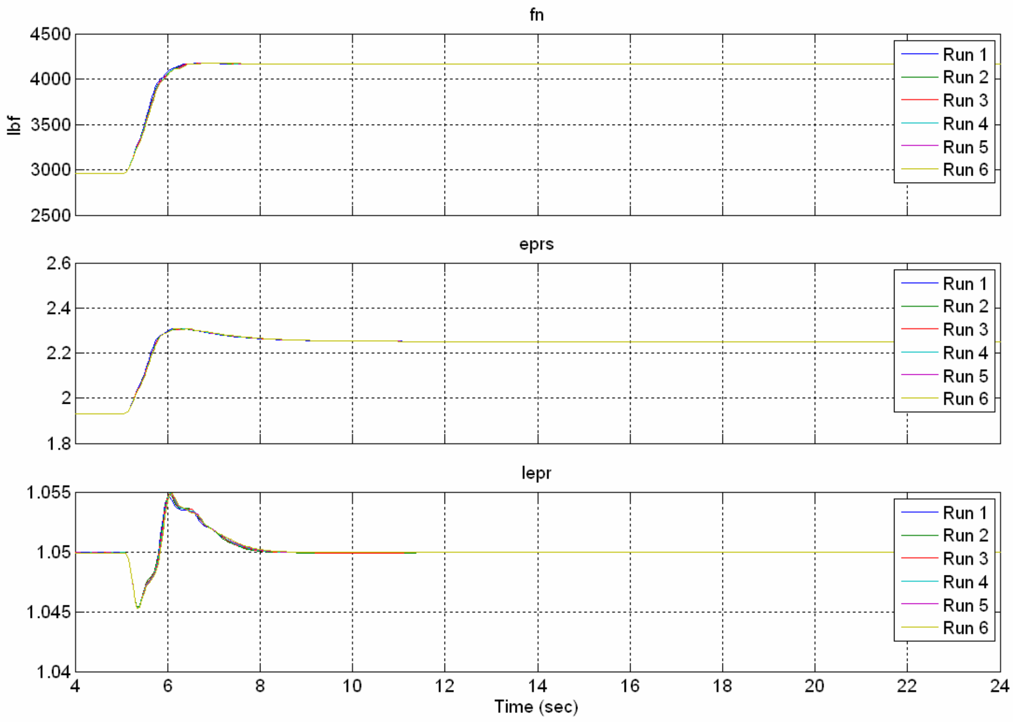


Figure 39: ADRC Controlled Variables at Operating Point #2

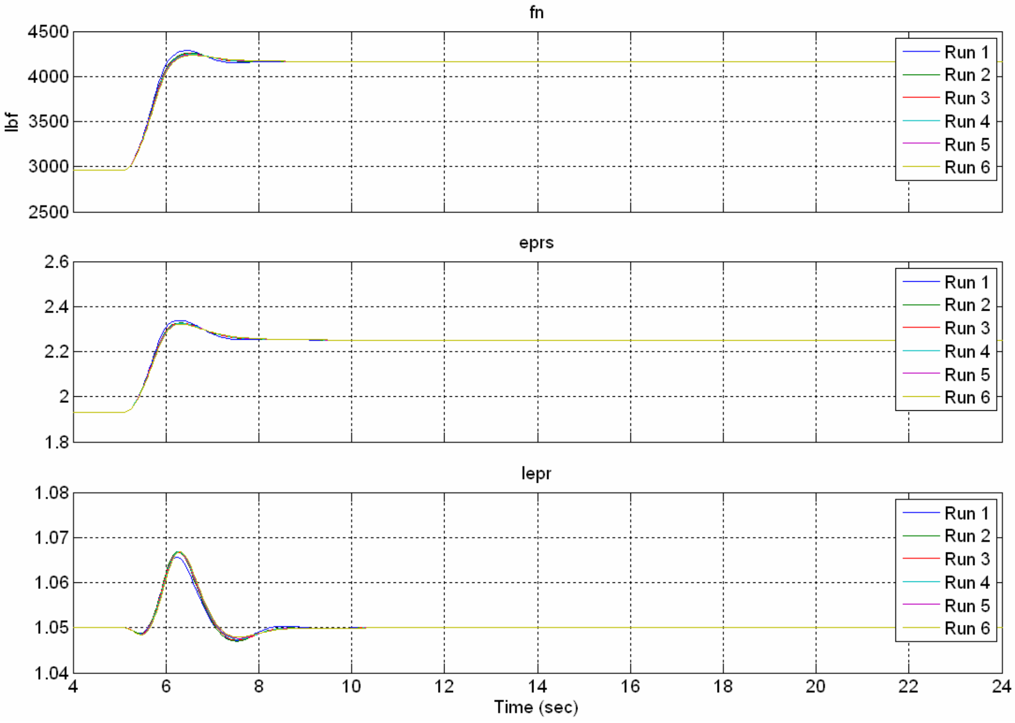


Figure 40: Nominal Controlled Variables at Operating Point #2

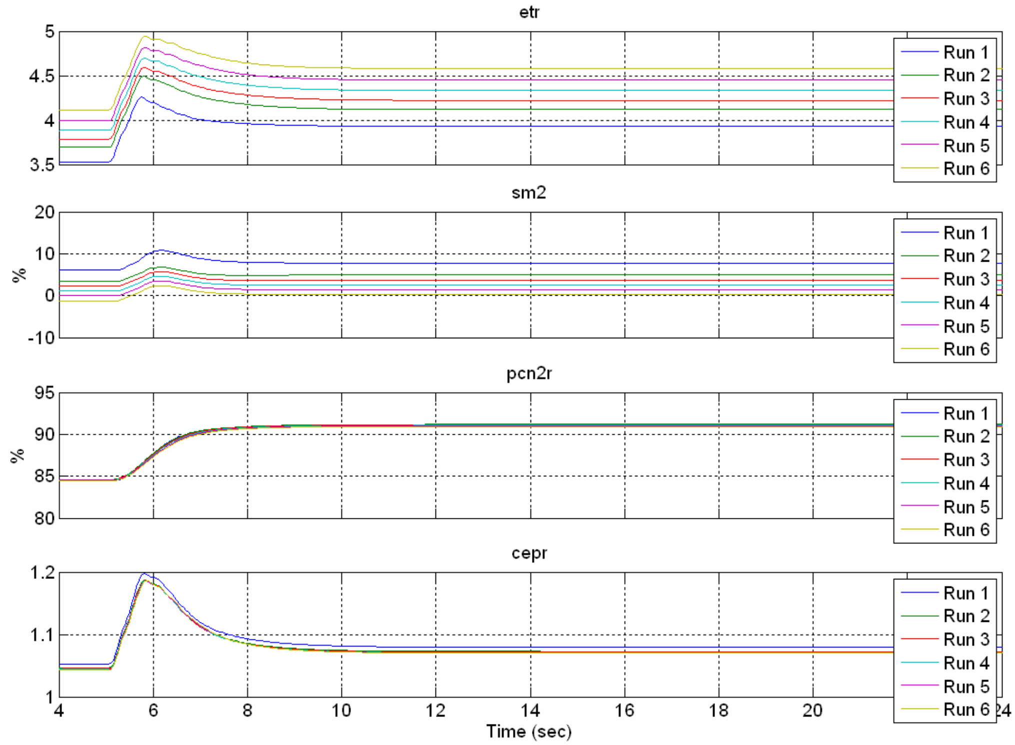


Figure 41: ADRC Uncontrolled Variables at Operating Point #2

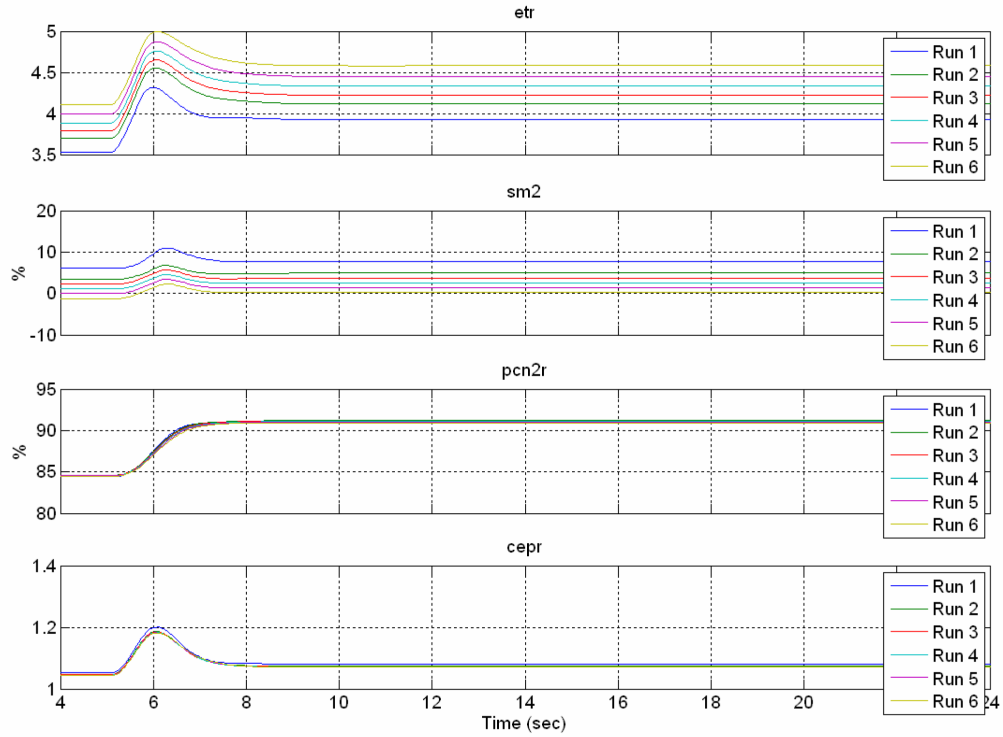


Figure 42: Nominal Uncontrolled Variables at Operating Point #2

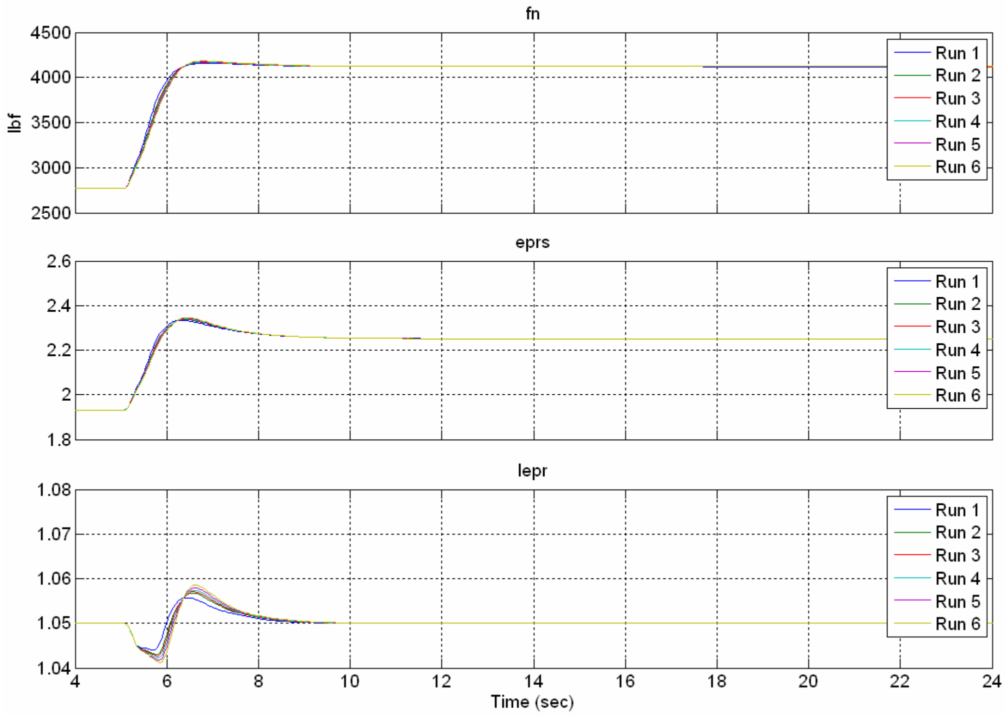


Figure 43: ADRC Controlled Variables at Operating Point #3

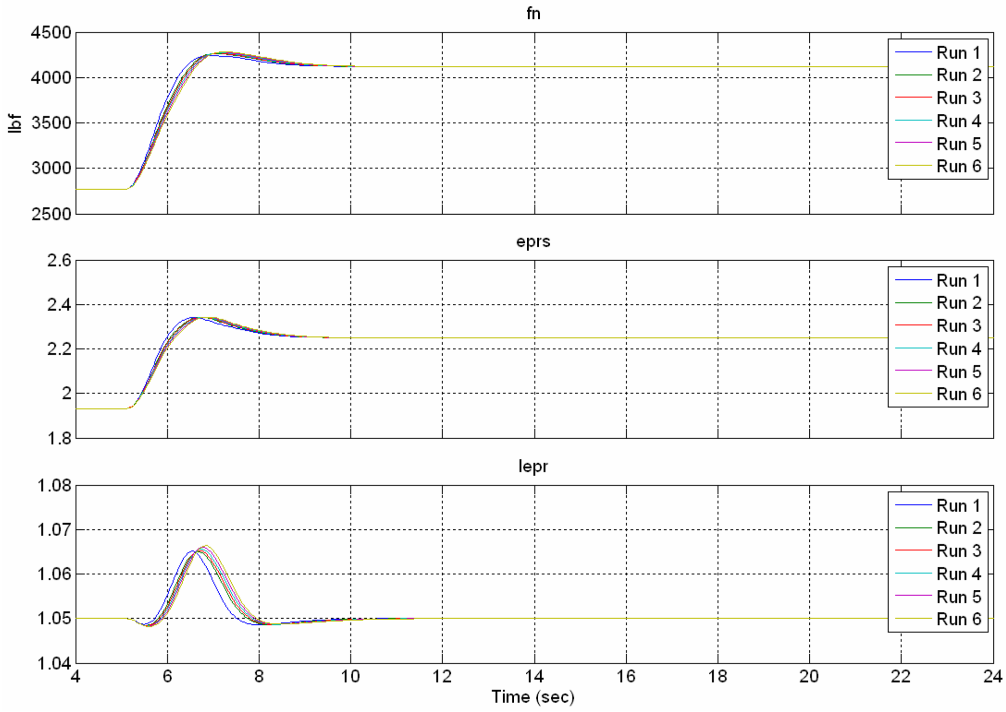


Figure 44: Nominal Controlled Variables at Operating Point #3

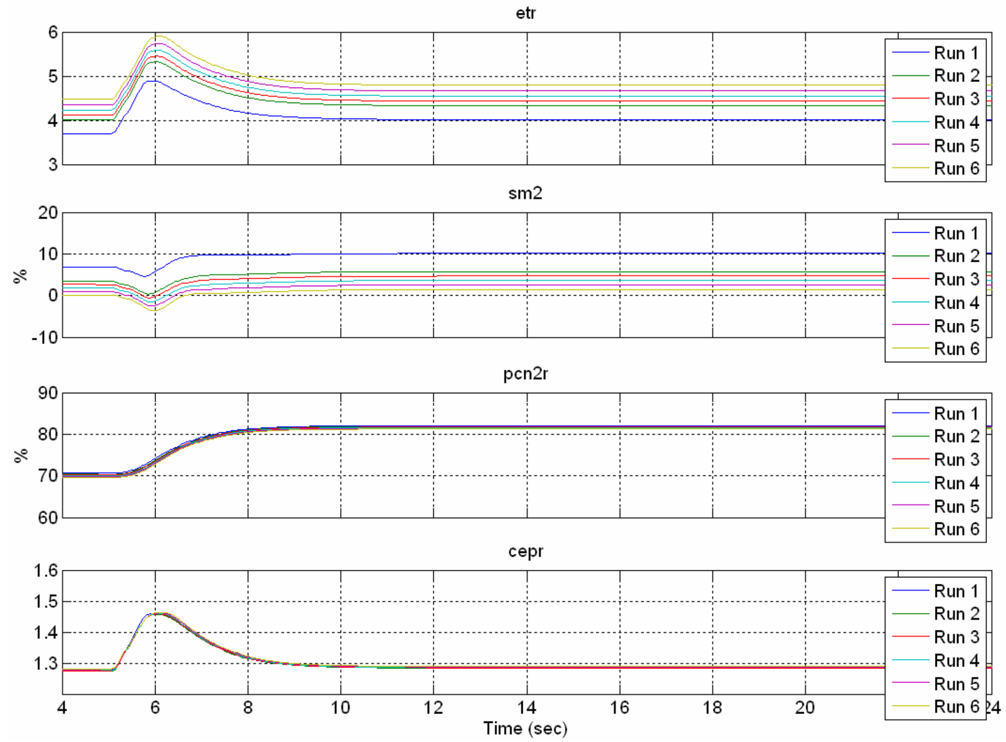


Figure 45: ADRC Uncontrolled Variables at Operating Point #3

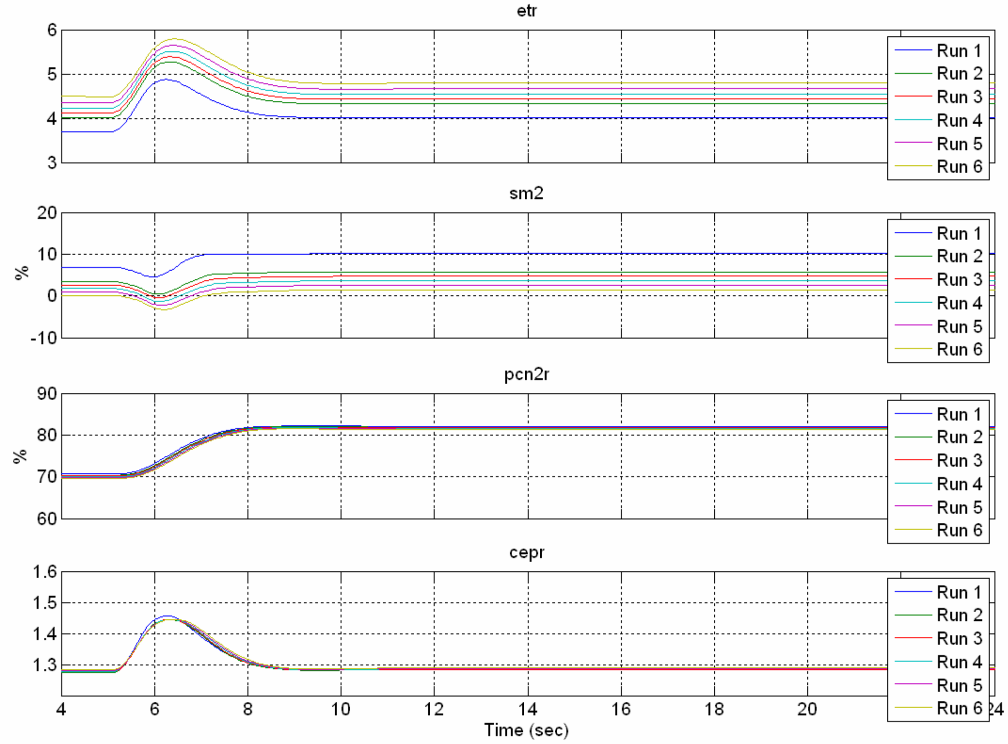


Figure 46: Nominal Uncontrolled Variables at Operating Point #3

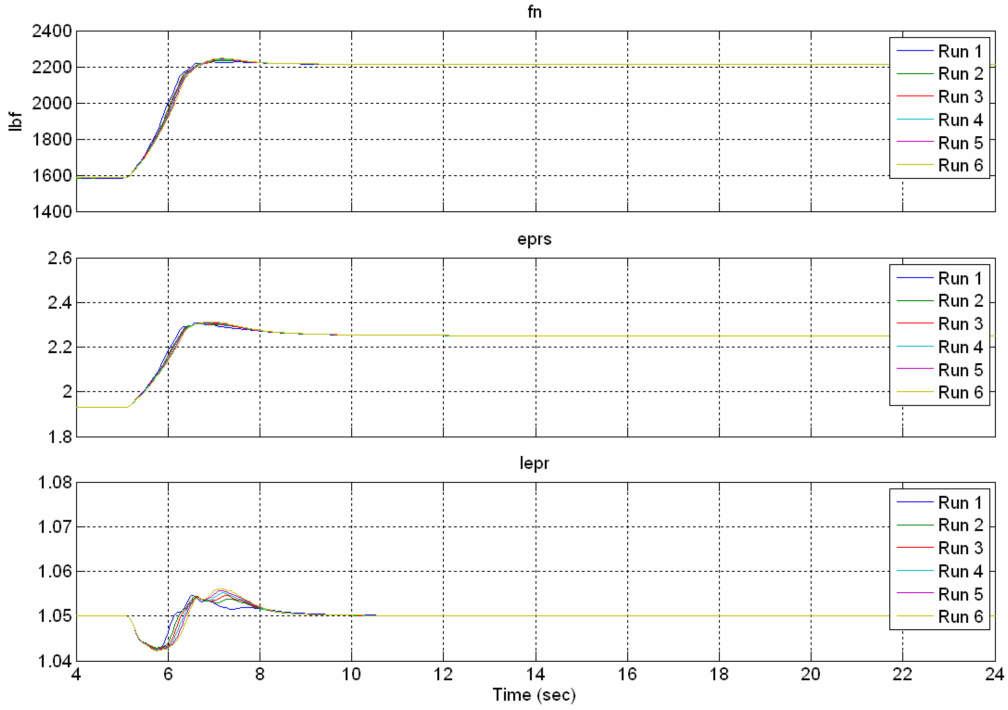


Figure 47: ADRC Controlled Variables at Operating Point #4

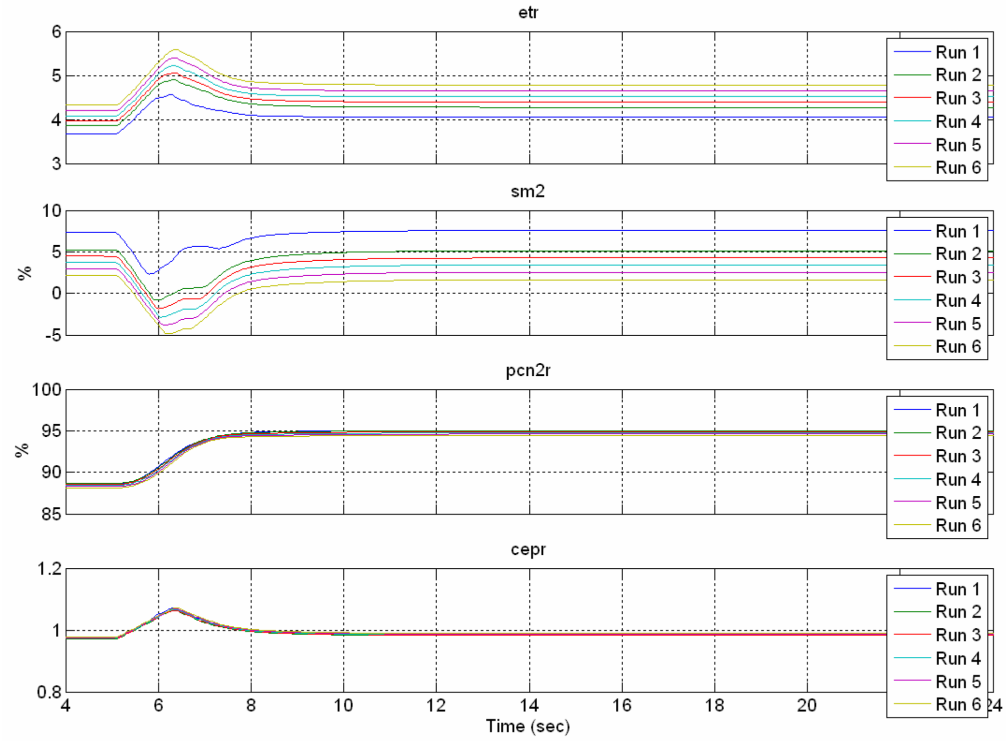


Figure 48: ADRC Uncontrolled Variables at Operating Point #4

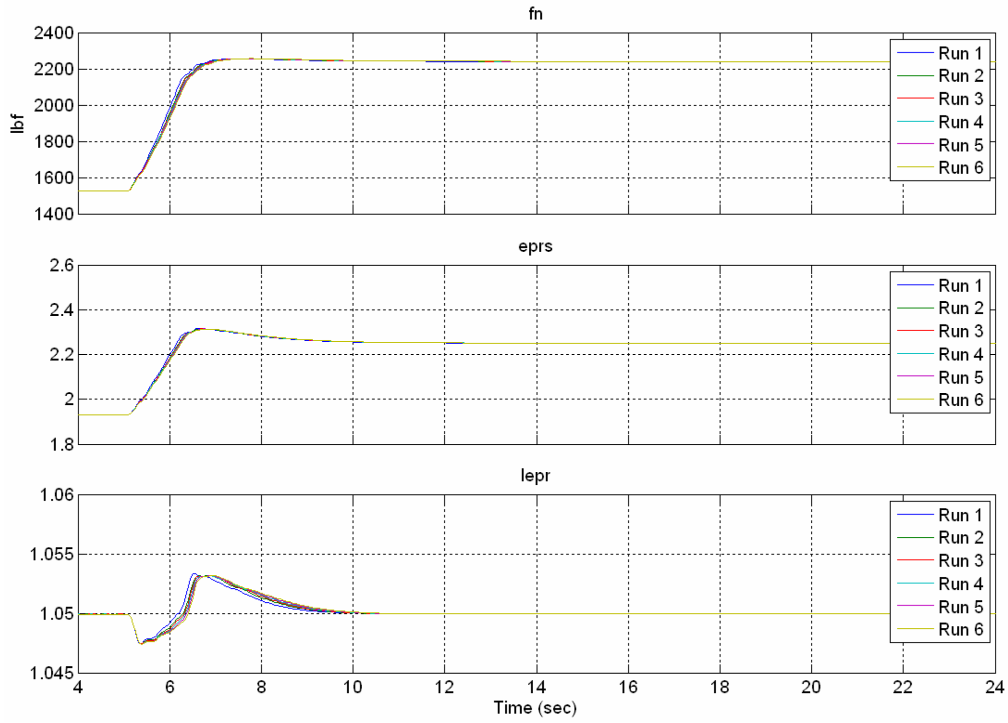


Figure 49: ADRC Controlled Variables at Operating Point #5

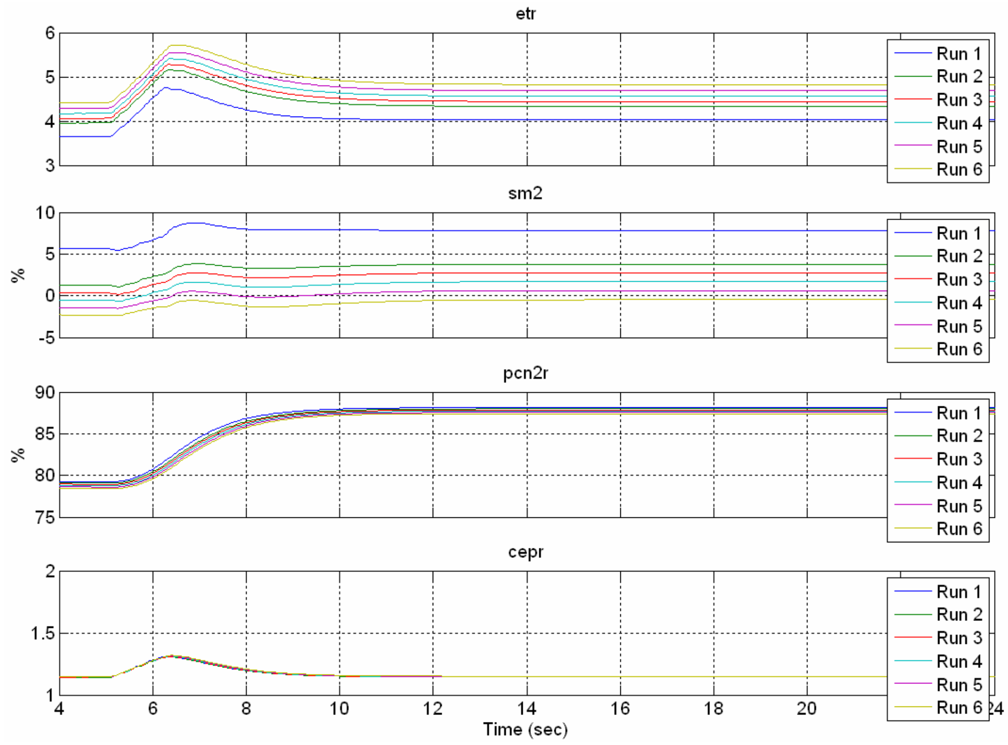


Figure 50: ADRC Uncontrolled Variables at Operating Point #5

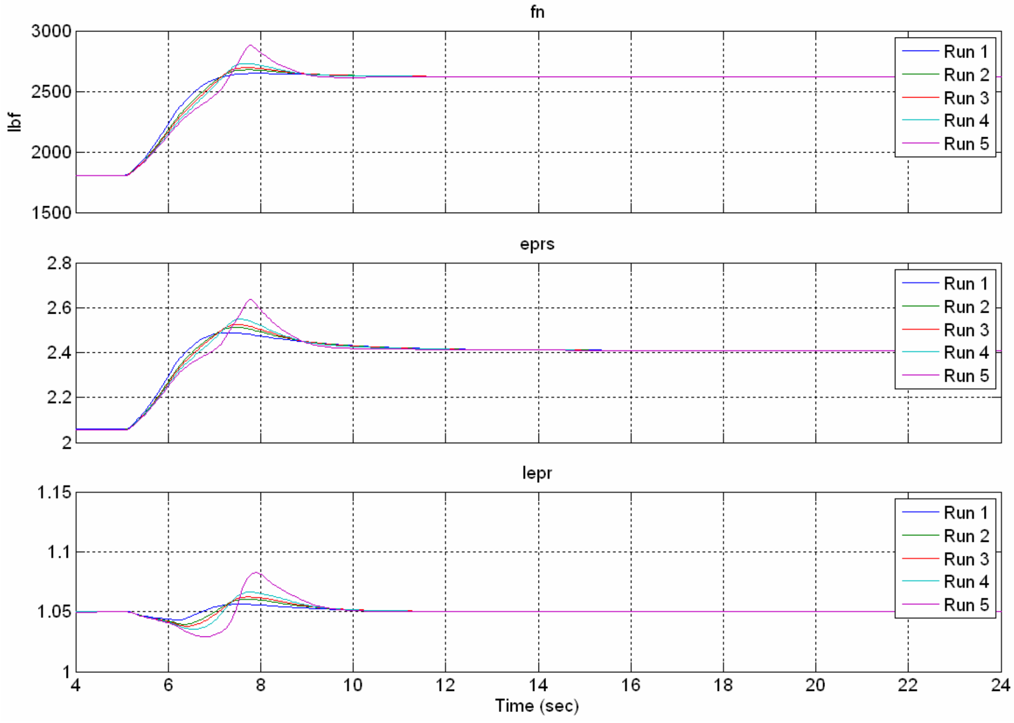


Figure 51: ADRC Controlled Variables at Operating Point #6

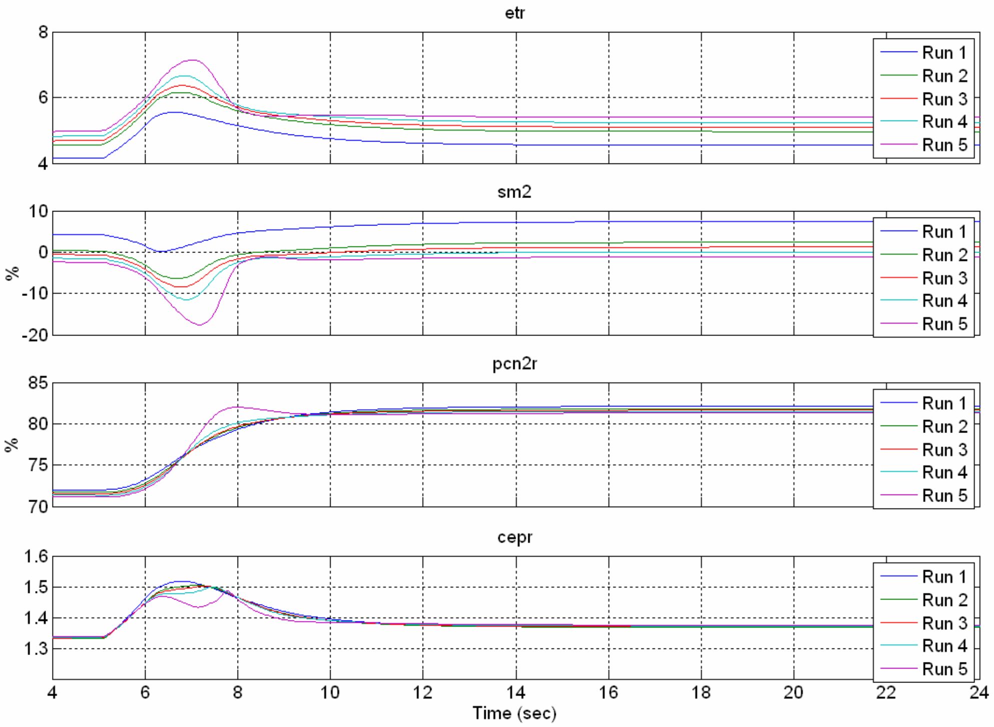


Figure 52: ADRC Uncontrolled Variables at Operating Point #6

Although the ADRC controller responded a bit faster and with less overshoot to the change in demand levels than the nominal controller, the real significance is in the simplicity of design of the new controller and how it was able to control engine thrust without being affected by degradation over a wide range of operation. The design procedure of the nominal controller basically involves running the CLM at several operating points to calculate a set of gains from Bode and Nyquist arrays at each operating point. The eighteen gains are each scheduled by six parameters, amounting to a total of 108 possible adjustments that can be made when configuring a single regulator on an actual engine. During the simulations, these gains change by as much as 200 percent.

In contrast, the five ADRC gains remained constant throughout all simulations.

$$\omega_c = 8, \quad \omega_o = 16, \quad B^{-1} = diag(.2, -.5, -.5) \tag{5.26}$$

There was no scheduling. Each gain was quickly tuned on the CLM just as if it would be on an actual engine. The engine was then simulated at multiple operating points to verify the performance of the new controller.

Signs of severe engine wear begin to show at test operating point #6. However, this flight condition takes the engine near the operability limit of the low speed regulator and subsonic range. Perhaps a DTOC controller or an ESO with two extended states may help in this situation, but it is to be expected that no electronics will fix severe physical wear on the engine components when they are placed under heavy stresses.

CHAPTER VI

CONCLUDING REMARKS

A new framework for controlling uncertain systems has been proposed. It is primarily composed of the ESO and control law. First, the ESO was discretized as a current discrete estimator for improved accuracy at lower sampling rates and extended for systems and disturbances of arbitrary order. Secondly, a GPID control law was proposed that fixes many of the problems that are currently associated with PID, allowing it to reject disturbances even without an ESO. The combination of GPID and ESO is also discussed. Many permutations exist. Next, the ADRC and GPID structures are extended to multivariable systems and a gain and order estimation algorithm is proposed. Finally, multivariable ADRC is applied to the turbofan engine for proof of concept.

From here, there are many research directions. This is only a beginning. For example, the possible applications consist of a number of new and existing configurations that may improve performance and simplify design.

1. Combine GPID and CDESO where m>0 and h=0.

2. Combine GPID and CDESO where m>0 and h>0, and $\hat{f}$ is not fed back.

3. Combine GPID and CDESO where m>0 and h>0, and $\hat{f}$ is fed back.

4. Combine GPID and CDESO where m=0 and h>0, and $\hat{f}$ is fed back. (Current ADRC)

5. Combine multivariable GPID and CDESO in any of the above configurations

6. Create a new tracking control structure based on GPID.

7. Apply GPID tuning rules and a prefilter $H = C_1C^{-1}$ to existing PID controllers.

8. Apply GPID tuning rules to existing I-PD controllers.

9. Apply GPID concepts to simplify design of controllers using the Integral Control structure.

10. Apply GPID concepts to simplify design of 2-DOF controllers.

More importantly, the next step after finding multiple solutions to any problem is to determine when to use each solution, first by experimental comparison and then by analytical rigor. Research should be conducted to find out what types of systems suit each solution best by establishing the limitations of each, i.e. stability and robustness. However, PID is used in 95% of all industrial applications today without this type of analysis.

The new framework provides a solution to a large class of problems that can be representing in terms of $Y^{(n)} = F + BU$. Time and time again, it has been shown experimentally that ADRC compensates for dynamic uncertainty as well as about 50% of uncertainty in system gain *B*. Strong nonlinearities in gain still need to be somewhat modeled or scheduled. Research should be conducted to investigate the limitations of the new framework before this type of scheduling is required. Important future work also includes controlling systems with more outputs than inputs. Another research possibility is that the framework may be extended to multi-order multivariable systems.

$$\begin{aligned} &y_1^{(n1)} = f_1 + b_1 U \\ &\vdots \\ &y_q^{(nq)} = f_q + b_q U \end{aligned} \tag{6.1}$$

Although the CDESO is stable at low sample rates under certain experimental conditions, stability with respect to sampling should be determined for the CDESO as well as for the whole control structure. A good starting point would be to determine stability for a known linear model by representing the structure as a 2-DOF controller.

As far as the jet engine is concerned, this is only the beginning. This research includes a first order multivariable ADRC controller that was digitized using Euler integration and a diagonal B^{-1} matrix was tuned by hand for the purpose of proof of concept. However, a number of issues arose that implied an even better solution may exist. First, the sampling time is very low in the engine controller, which is a good situation for *q*-parallel CDESO observer loops instead of Euler integration. Secondly, the tuning of ω_o and ω_c was very tight, meaning that multiple extended states could be used to increase the slope at the cutoff frequency and allow for more bandwidth. Furthermore, the simplest possible control law was used in the simulations. The use of more advanced control laws, such as a tracking controller, non-smooth functions, [21], [51], [62] or recent discrete time-optimal techniques [27], [28] are promising options for further improvements in performance.

Another possibility is to assume the system is of higher order. The method proposed to determine system order and gain can be adapted to deal with systems not initially at rest, yet begin in a steady state condition. It can then be used in a preliminary test, moving research one step closer towards the realization of a truly universal “black-box” controller or at least a closed loop estimation of *B* while the system is being controlled and subjected to disturbances.

Where modern multivariable control schemes are limited, this approach appears well suited for complex nonlinear systems with incomplete model information. The ultimate goal is to offer a degree of manual adjustment to account for variations between engines without sacrificing performance, while being robust enough to withstand slow degradations from aging or damage. Preliminary results of these simulation tests on a rather complex turbofan model are a good indication to the power of the dynamic decoupling method proposed here. Mathematical models are often inaccurate when representing nonlinear multivariable systems. Gain scheduling helps in this area, but makes tuning even worse than it was before. The information needed to effectively control a physical system can be extracted from input-output data in real time and used in the control law without an explicit mathematical model. With this method, the information required of a system is its order, i.e. how many integrators there are, and its high frequency gain characteristics.

REFERENCES

1. Pratt & Whitney Aircraft, "The Aircraft Gas Turbine Engine and Its Operation," United Technologies Corporation, December 1982.

2. J.D. Mattingly, *Elements of Gas Turbine Propulsion*, McGraw-Hill, Inc, 1996.

3. B.R. Endurthi, "Linearization and health estimation of a turbofan engine," Master's thesis, Cleveland State University, December 2004.

4. A. Kreiner, and K. Lietzau, "The use of onboard real-time models for jet engine control," MTU Aero Engines, Germany, 2003.

5. S. Adibhatla and Z. Gastineau, "Tracking filter selection and control mode selection for model based control," *AIAA 94-3204,* 30th Joint Propulsion Conference and Exhibit, Indianapolis, IN, June 1994.

6. K.I. Parker and T.-H. Guo, "Development of a turbofan engine simulation in a graphical simulation environment," *NASA/TM-2003-212543*, August 2003.

7. M. French, "Development of a compact real-time turbofan engine dynamic simulation," SAE Paper No. 821401, Aerospace Congress & Exposition, Anaheim, CA, October 25-28, 1982.

8. J.A. Polley, S. Adibhatla, and P.J. Hoffman, "Multivariable turbofan engine control for full flight operation," Gas Turbine and Aeroengine Congress and Exposition, Amsterdam, The Netherlands, June 5-8, 1988.

9. S. Adibhatla and K.L. Johnson, "Evaluation of a nonlinear PSC algorithm on a variable cycle engine," AIAA/SAE/ASME/ASEE 29th Joint Propulsion Conference and Exhibit, Monterey, CA, June 38-30, 1993.

10. M.P. Boyce, *Gas Turbine Engineering Handbook, Second Edition*, Butterworth-Heinemann, 2002.

11. N. Cumpsty, *Jet Propulsion: A Simple Guide*, Cambridge University Press, 2002.

12. D. Mattern and S. Garg, “Propulsion system performance resulting from an integrated flight/propulsion control design,” AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, August 10-12, 1992.

13. S.R. Watts and S. Garg, “A comparison of multivariable control design techniques for a turbofan engine control,” International Gas Turbine and Aeroengine Congress and Exposition, Houston, TX, June 5-8, 1995.

14. S. Garg, “A simplified scheme for scheduling multivariable controllers,” *IEEE Control Systems*, August 1997.

15. D.K. Frederick, S. Garg and S. Adibhatla, “Turbofan engine control design using robust multivariable control technologies,” IEEE Transactions on Control Systems Technology, November 2000.

16. S.R. Watts and S. Garg, “An optimized integrator windup protection technique applied to a turbofan engine control,” AIAA Guidance Navigation and Control Conference, San Diego CA, July 29-31, 1996.

17. S. Adibhatla, G.J. Collier, X. Zhao, and D. Frederick, “Advanced multivariable technology for engine controls,” NASA Lewis Large Engine Technology, 1998.

18. R.L. DeHoff, W.E. Hall, Jr., R.J. Adams, and N.K. Gupta, “F100 Multivariable Control Synthesis Program, Volume I – Development of F100 Control System,” Aeronautical & Marine Systems Sector, Palo Alto, CA, June 1977.

19. S. Adibhatla, P. Cooker, A. Pajakowski, and B. Romine, “STOVL Controls Technology, Volume II – F110/STOVL Propulsion System Description,” NASA Lewis Research Center, July 1994.

20. J.M. Edmunds, “Control system design using closed-loop Nyquist and Bode arrays,” *International Journal on Control,* Vol. 30, No. 5, 1979, pp. 773-802.

21. Z. Gao, Y. Huang and J. Han, “An alternative paradigm for control system design,” *Proc. of the 40th IEEE Conference on Decision and Control*, December 2001, pp. 4578-4585.

22. Z. Gao, “From linear to nonlinear control means: A practical progression,” *ISA Transactions*, 2001.

23. J. Zeller, M. Zhu, T. Stimac, and Z. Gao, "Nonlinear digital control implementation for a DC-DC power converter," 36th Intersociety Energy Conversion Engineering Conference, July 29-August 2, 2001.

24. Y. Hou, Z. Gao, F. Jiang, and B. T. Boulter, "Active disturbance rejection control for web tension regulation," IEEE Conference on Decision and Control, 2001.

25. F. Jiang and Z. Gao, "An application of nonlinear PID control to a class of truck ABS problems," IEEE Conference on Decision and Control, 2001.

26. F. Jiang and Z. Gao, "An adaptive nonlinear filter approach to vehicle velocity estimation for ABS," *Proc. of the 2000 IEEE International Conference on Control Applications*, Anchorage, AK, September 25-27, 2000, pp.490-495.

27. Z. Gao, "On discrete time optimal control: A closed form solution," *Proc. of the 2004 American Control Conference*, Boston MA, June 30-July 2, 2004, pp. 52-58.

28. Z. Gao and S. Hu, "On properties and applications of a new form of discrete time optimal control law," 2004 IAS Annual Meeting, Seattle WA, October 3-7, 2004.

29. N. Minorsky, "Directional stability and automatically steered bodies," *Journal of the American Society of Naval Engineering*, Vol. 34, 1922, p.280.

30. K. Astrom and T. Hagglund, "PID Control," *The Control Handbook*, W. S. Levine, Ed., CRC Press and IEEE Press, 1996, p. 198.

31. K.L. Chien, J.A. Holmes, and J.B. Reswick, "On the automatic control of generalized passive systems," *Transactions of the ASME*, Vol. 74, 1952, pp. 175-185.

32. K. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*. North Carolina: Instrument Society of America, 1995, pp. 74, 11-116, 128, 134-156, 160-162.

33. J. T. Gillis, "State space," *The Control Handbook*, W. S. Levine, Ed., CRC Press and IEEE Press, 1996, pp. 80-81.

34. J.G. Ziegler and N.B. Nichols, “Optimal settings for automatic controllers,” *Transactions of the ASME*, Vol. 64, 1942, pp. 759-768.

35. R. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, Series D, 1960, pp. 35–45.

36. D. Luenberger, “Observers for multivariable systems,” *IEEE Transactions on Automatic Control*, Vol. 11, No. 2, 1966, pp. 190–197.

37. J. C. Doyle and G. Stein, “Multivariable feedback design: Concepts for a classical/modern synthesis,” *IEEE Transactions on Automatic Control*, Vol. 26, No. 1, 1981, pp. 4 –16.

38. B. Friedland, *The Control Handbook*, CRC Press, IEEE Press, 1996, pp. 607–618.

39. C.-T. Chen, *Linear System Theory and Design* (Oxford Series in Electrical and Computer Engineering), 3rd ed. Oxford University Press, August 1998.

40. G.F. Franklin, J.D. Powell, and M. Workman, *Digital Control of Dynamic Systems,* 3rd ed., Menlo Park, CA: Addison Wesley Longman, Inc., 1998, pp. 49-50, 322-325, 328-337.

41. G.F. Franklin, J.D. Powell, and A. Emami-Naeni, *Feedback Control of Dynamic Systems,* 4th ed., Upper Saddle River, NJ: Prentice-Hall, Inc., 2002, pp. 239-242, 601-604.

42. A. Radke and Z. Gao, “A survey of state and disturbance observers for practitioners,” *Proc. of the American Control Conference*, Minneapolis, MN, June 14-16, 2006.

43. J. Liu, “Comparative study of differentiation and integration techniques for feedback control systems,” Master’s thesis, Cleveland State University, December 2002.

44. H. R. Simson, “Performance measures and optimization condition for a third-order sampled-data tracker,” *IEEE Transactions on Automatic Control*, 1963, pp. 182–183.

45. D. Tenne and T. Singh, “Characterizing performance of $\alpha - \beta - \gamma$ filters,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 3, July 2002, pp. 1072–1087.

46. G. Basile and G. Marro, “On the observability of linear, time-invariant systems with unknown inputs,” *Journal of Optimization Theory and Applications*, Vol. 2, No. 6, 1969, pp. 410–415.

47. C. Johnson, “Accomodation of external disturbances in linear regulator and servomechanism problems,” *IEEE Transactions on Automatic Control*, Vol. AC-16, No.6, December 1971, pp. 635-644.

48. G.H. Hostetter and J. Meditch, “On the generalization of observers to systems with unmeasurable, unknown inputs,” *Automatica*, Vol. 9, 1973, pp. 721–724.

49. C. Johnson, “Theory of disturbance-accommodating controllers,” *Control and Dynamic Systems*, Vol. 12, 1976, pp. 387–489.

50. T. Umeno and Y. Hori, “Robust speed control of DC servomotors using modern two degrees-of-freedom controller design,” *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 5, October 1991, pp. 363–368.

51. J. Han, “Nonlinear design methods for control systems,” Proc. 14th IFAC World Congress, Beijing, 1999.

52. Z. Gao, “Scaling and bandwidth-parameterization based controller tuning,” *American Control Conference*, June 2003, pp. 4989 – 4996.

53. Y. Hori, K. Shimura, and M. Tomizuka, “Position/force control of multi-axis robot manipulator based on the TDOF robust servo controller for each joint,” *Proc. of the American Control Conference ACC/WM9*, Vol. 1, 1992, pp. 753-757.

54. E. Schrijver and J. van Dijk, “Disturbance observers for rigid mechanical systems: equivalence, stability, and design,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 124, No. 4, 2002, pp. 539–548.

55. J.B. Burl, *Linear Optimal Control: H2 and H∞ methods*, Menlo Park, CA: Addison Wesley Longman, Inc., 1999, pp. 308-314.

56. R. Bickel and M. Tomizuka, “Passivity-based versus disturbance observer based robot control: equivalence and stability,” *Journal of Dynamic Systems, Measurement, and Control*, Vol. 121, March 1999, pp. 41-47.

57. S.J. Kwon and W.K. Chung, “Robust performance of the multiloop perturbation compensator,” *IEEE/ASME Transactions on Mechatronics*, Vol. 7, No. 2, June 2002, pp.190-200.

58. Z. Gao, S. Hu, and F. Jiang, “A novel motion control design approach based on active disturbance rejection,” Proc. of the 40th IEEE conference on Decision and Control, December 2001, p. 4974.

59. B. Sun and Z. Gao, “A DSP-based active disturbance rejection control design for a 1KW H-Bridge DC-DC power converter,” *IEEE Transactions on Industrial Electronics*, 2004.

60. R. Kotina, Z. Gao, and A. J. van den Bogert, “Modeling and control of human postural sway,” XXth Congress of the International Society of Biomechanics, Cleveland, Ohio, July 31 - August 5, 2005.

61. F. Goforth, “On motion control design and tuning techniques,” Proc. of the American Control Conference, Boston, MA, June 30 – July 2, 2004.

62. Y. Huang, K. Xu, J. Han, and J. Lam, “Flight control design using extended state observer and non-smooth feedback ,” TuA08-3, Proc. of 40th IEEE Conference on Decision and Control, Orlando, FL, December 4-7, 2001.

63. R. Miklosovic, A. Radke, and Z. Gao, “Discrete implementation and generalization of the extended state observer,” Proc. of the American Control Conference, Minneapolis, MN, June 14-16, 2006.

64. R. Miklosovic and Z. Gao, “A dynamic decoupling method for controlling high performance turbofan engines,” Proc. of the 16th IFAC World Congress, Prague, Czech Republic, July 4-8, 2005.

65. R. Miklosovic and Z. Gao, “A robust two-degree-of-freedom control design technique and its practical application,” Proc. of the IEEE/IAS World Conference, Seattle, WA, Oct. 3-7, 2004.

66. B.K. Kim, H.-T. Choi, W.K. Chung and I.H. Suh, “Analysis and design of robust motion controllers in the unified framework,” *Journal of Dynamic Systems, Measurement, and Control: ASME*, Vol. 124, June 2002, pp. 313-321.

67. J.A. Profeta III, W.G. Vogt, and M.H. Mickle, “Disturbance estimation and compensation in linear systems,” *IEEE Trans. Aerospace and Electronic Systems*, Vol. 26, No. 2, March 1990, pp. 225-231.

68. B. Francis and W. Wonham, “The Internal Model Principal of Control Theory,” *Automatica*, Vol. 12, 1976, pp. 457-465.

69. S. Endo, H. Kobayashi, C.J. Kempf, S. Kobayashi, M. Tomizuka, and Y. Hori, “Robust digital tracking controller design for high-speed positioning systems,” *Control Engineering Practice*, Vol. 4, No. 4, 1996, pp. 527-536.

70. A. Tesfaye, H.S. Lee and M. Tomizuka, “A sensitivity optimization approach to design of a disturbance observer in digital motion control.” *IEEE/ASME Transactions on Mechatronics*, Vol. 5, No. 1, March 2000, pp. 32-38.

71. R.H. Bishop and R.C. Dorf, “The Routh-Hurwitz Stability Criterion” *The Control Handbook*, W. S. Levine, Ed., CRC Press and IEEE Press, 1996, pp. 131-133.

72. N.S. Nise, *Control Systems Engineering*. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc., 1992, pp. 277-301.

73. T. Ren, “Applications of active disturbance rejection and a new parameter estimation method,” Master’s thesis, Department of Electrical and Computer Engineering, Cleveland State University, December 2004.

74. E. C. Ifeachor, B. W. Jervis, *Digital Signal Processing, A Practical Approach*, 2nd Ed., Prentice Hall, 2001.

75. T.T. Hartley, G.O. Beale, and S.P. Chicatelli, *Digital Simulation of Dynamic Systems: A Control Theory Approach*, New Jersey: Prentice Hall Inc., 1994.

76. G.P. Sallee, “Performance deterioration based on existing (historical) data; JT9D jet engine diagnostics program,” NASA Contractor Report 135448, 1978.

77. J.S. Litt, K.I. Parker, and S. Chatterjee, “Adaptive gas turbine engine control for deterioration compensation due to aging,” *16th International Symposium on Air Breathing Engines*, ISABE 2003-1056, Cleveland, OH, August 31 – September 5, 2003.

78. J.S. Litt and E.M. Aylward, “Adaptive detuning of a multivariable controller in response to turbofan engine degradation,” *NASA/TM – 2003-212723*, October 2003.

79. L.A. Urban, “Gas Path Analysis Applied to Turbine Engine Condition Monitoring,” AIAA/SAE 8th Joint Propulsion Specialist Conference, November 29-December 1, 1972, New Orleans, LA, AIAA no. 72-1082.

80. O. Sasahara, “JT9D engine/module performance deterioration results from back to back testing,” ISABE 85-7061, Seventh International Symposium on Air Breathing Engines, Beijing, PRC, September 2-6, 1985.

81. D. Simon and D.L. Simon, “Aircraft turbofan engine health estimation using constrained Kalman filtering,” ASME-GT2003-38584, International Gas Turbine and Aeroengine Congress and Exposition, Atlanta, GA, June 2003.