Rochester Institute of Technology

RIT Scholar Works

Theses

5-4-2009

Feasibility assessment of a Kalman filter approach to fault detection and fault-tolerance in a highly unstable system: The RIT heart pump

Erin Gillespie

Follow this and additional works at: <https://scholarworks.rit.edu/theses>

Recommended Citation

Gillespie, Erin, "Feasibility assessment of a Kalman filter approach to fault detection and fault-tolerance in a highly unstable system: The RIT heart pump" (2009). Thesis. Rochester Institute of Technology.
Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

Feasibility Assessment of a Kalman Filter Approach to Fault Detection and Fault-Tolerance in a Highly Unstable System: The RIT Heart Pump

By

Erin Gillespie

A Thesis submitted in Partial Fulfillment
of the Requirement for the

**MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING**

Approved By:

Dr. Wayne W. Walter
Department of Mechanical Engineering

Dr. Agamemnon L. Crassidis
Department of Mechanical Engineering

Dr. Margaret Bailey
Department of Mechanical Engineering

Dr. Steven Day
Department of Mechanical Engineering

Dr. Edward C. Hensel
Department Head of Mechanical Engineering

**DEPARTMENT OF MECHANICAL ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY**

May 4, 2009

REPRODUCTION PERMISSION STATEMENT

Permission Granted:

Position Sensor Fault Detection and Fault Tolerant Operation of the RIT Heart Pump

I, Erin Gillespie, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: _____

Signature of Author: _____

Abstract

The purpose of this project is to assess the feasibility of a Kalman Filter approach for fault detection in a highly unstable system, specifically the heart pump currently under development at RIT. Simulations and experimental work were completed to determine the effects of possible position sensor fault conditions on the system; that information was then used in conjunction with a pair of Kalman filters to create a method of detecting faults and providing fault-tolerant operation. The heart pump system was modeled using Simulink and then the fault diagnosis and tolerance system was added to the model and tested via simulation in SIMULINK™. The simulations showed the filters were able to calculate and remove bias caused by any type of position sensor error, provided the estimated plant model is nearly identical to the actual plant model. Sensitivity analysis showed that the fault detection/fault-tolerance method is extremely sensitive to discrepancies between the estimated plant model and actual pump behavior. Because of this, it is considered unfeasible for implementation on a real system. Experimental results confirmed these findings, demonstrating the drawbacks of model-based fault detection and tolerance methods.

Acknowledgments

Special thanks to Dr. Wayne Walter, Dr. Steven Day, Dr. Margaret Bailey, Dr. Agamemnon Crassidis, Dr. Mark Olles, Dr. Edward Hensel, and the students in the RIT heart pump lab for their help and guidance with this project. Thanks also to Dr. John Crassidis and Yang Cheng at the University of Buffalo for their assistance.

In addition, I would like to thank my friends and family, especially Scott, for their support and encouragement during the completion of this thesis.

Table of Contents

Acknowledgments.....	iii
List of Tables	v
List of Figures	vi
Variables	x
Abbreviations.....	x
1.0 Introduction.....	1
1.1 Heart Pumps/RIT Heart Pump Project.....	1
1.2 Assessment of Need.....	3
1.3 History.....	3
1.3.1 Fault Detection and Fault-Tolerance	4
1.3.2 Heart Pumps and Magnetically Levitated Bearings	9
1.3.3 Control of Magnetically Levitated Bearings	10
1.4 Scope.....	12
2.0 Heart Pump	13
2.1 Hardware and Configuration.....	13
2.2 Controller	16
2.3 System Model	16
3.0 Fault Detection and Fault Tolerance Method Selection	20
3.1 Needs.....	20
3.2 Selection.....	20
3.3 Kalman Filter	21
3.4 Multiple Model Adaptive Estimation	24
3.5 Sensor Drift Rate Estimation	27
3.6 Covariance Analysis	29
3.7 Design	29
4.0 Testing.....	34
4.1 Simulation.....	34
4.1.1 Results	36
4.1.2 Sensitivity to Plant Model Discrepancies.....	58
4.2 Experimental Testing.....	59
4.2.1 Test Rig	60
4.2.2 Preliminary Experimental Test Results	60
4.2.3 LVADsim35 Results	65
5.0 Interpretation of Results.....	66
6.0 Conclusions.....	68
7.0 Future Recommendations	69
7.0 References.....	70
Appendices.....	75
Appendix A: Simulink Models and Associated Filter Outputs	75
A.1: Initial Model with Gain Calculation Kalman Filter, and Results.....	75
A.2: Four-State Model with Gain Calculation Kalman Filter, Covariance Analysis, and Feedback Switch, and Results	89
Appendix B: Individual Sensor Equations.....	100
Appendix C: Position Sensor Fault Emulation Test Procedure	101
Appendix D: Plant Model Derivation	103

List of Tables

Table 1: Fault Modes for an Active Magnetic Bearing System [6] 4

Table 2: Considered Fault Types and Consequences 35

Table 3: Fault Conditions.....35

List of Figures

Figure 1: Implanted LVAD [1]	2
Figure 2: Heart Pump Assembly View [1]	13
Figure 3: Complete LVAD System [1]	14
Figure 4: Rotor Magnet Polarity and Arrangement [1]	14
Figure 5: Impeller [1]	14
Figure 6: Sensor Arrangement, End View of a HESA sensor installed in the heart pump	15
Figure 7: Diagram of signal path from the HESA sensors to the pump	15
Figure 8: PID Controller	16
Figure 9: Basic Simulink Model. ‘u1’-‘u4’ are controller output values. The dead zone block limits the controller force to better mimic the actual system limitations.....	17
Figure 10: Illustration of rotor with coordinate system	19
Figure 11: Illustration of Kalman Filter Functionality [25]	23
Figure 12: Final Simulink Model	33
Figure 13: Error Block	36
Figure 14: Control Effort for y , θ_x , for System Operating with No Position Sensor Errors ..	36
Figure 15: Control Effort for x , θ_y , for System Operating with No Position Sensor Errors ..	37
Figure 16: Position and Velocity Output for y , θ_x , for System Operating with No Position Sensor Errors	37
Figure 17: Position and Velocity Output for x , θ_y , for System Operating with No Position Sensor Errors	38
Figure 18: Bias Calculation Filter Output for y , θ_x , for System Operating with No Position Sensor Errors	39
Figure 19: Bias Calculation Filter Output for x , θ_y , for System Operating with No Position Sensor Errors	39
Figure 20: Control Effort for y , θ_x , for Error Condition 1, starting at $t=1s$	40
Figure 21: Control Effort for x , θ_y , for Error Condition 1, starting at $t=1s$	40
Figure 22: Position and Velocity Output for y , θ_x , for Error Condition 1, starting at $t=1s$	41
Figure 23: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1s$	42
Figure 24: Position and Velocity Output for x , θ_y , for Error Condition 1, starting at $t=1s$	42
Figure 25: Bias Calculation Filter Output for y , θ_x , for Error Condition 1, starting at $t=1s$ ..	43
Figure 26: Bias Calculation Filter Output for x , θ_y , for Error Condition 1, starting at $t=1s$ ..	44
Figure 27: Control Effort for y , θ_x , for Error Condition 2, starting at $t=1s$	44
Figure 28: Control Effort for x , θ_y , for Error Condition 2, starting at $t=1s$	45
Figure 29: Position and Velocity Output for y , θ_x , for Error Condition 2, starting at $t=1s$	45
Figure 30: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 2, starting at $t=1s$	46
Figure 31: Position and Velocity Output for x , θ_y , for Error Condition 2, starting at $t=1s$	46
Figure 32: Bias Calculation Filter Output for y , θ_x , for Error Condition 2, starting at $t=1s$ ..	47
Figure 33: Bias Calculation Filter Output for x , θ_y , for Error Condition 2, starting at $t=1s$ ..	48
Figure 34: Control Effort for y , θ_x , for Error Condition 3, starting at $t=1s$	48
Figure 35: Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1s$	49
Figure 36: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1s$	49

Figure 37: Bias Calculation Filter Output for y , θx , for Error Condition 3, starting at $t=1s$ ..	50
Figure 38: Zoomed View of Bias Calculation Filter Output for y , θx , for Error Condition 3, starting at $t=1s$	50
Figure 39: Control Effort for y , θx , for Error Condition 4, starting at $t=1s$	51
Figure 40: Position and Velocity Output for y , θx , for Error Condition 4, starting at $t=1s$	51
Figure 41: Bias Calculation Filter Output for y , θx , for Error Condition 4, starting at $t=1s$ ..	52
Figure 42: Control Effort for y , θx , for Error Condition 17, starting at $t=1s$	53
Figure 43: Position and Velocity Output for y , θx , for Error Condition 17, starting at $t=1s$ ..	53
Figure 44: Bias Calculation Filter Output for y , θx , for Error Condition 17, starting at $t=1s$	54
Figure 45: Control Effort for y , θx , with low frequency sinusoidal noise.....	55
Figure 46: Position and Velocity Output for y , θx , with low frequency sinusoidal noise.....	55
Figure 47: Bias Calculation Filter Output for y , θx , with low frequency sinusoidal noise	56
Figure 48: Control Effort for y , θx , with high frequency sinusoidal noise.....	57
Figure 49: Position and Velocity Output for y , θx , with high frequency sinusoidal noise.....	57
Figure 50: Bias Calculation Filter Output for y , θx , with low frequency sinusoidal noise	58
Figure 51: AMB spring constant adjusted by 5%, no error conditions.	59
Figure 52: Levitation Test Rig Set-Up used for preliminary tests [1]	60
Figure 53: Nominal Pump Behavior	61
Figure 54: Rear positive y sensor unplugged (fault condition 11) experimental data	62
Figure 55: Rear positive x sensor shorted to +5V signal (error condition 27)	62
Figure 56: Front positive x sensor unplugged (error condition 20)	63
Figure 57: Observer Output	64
Figure 58: Observer Output	64
Figure 59: Rotor position output.....	65
Figure 60: y -position output from heart-pump emulator during normal pump operation. The output is essentially the same for the other position values.....	65
Figure 61: Fault detection output from heart-pump emulator during normal pump operation.	66
Figure 62: Position and Velocity Output: No Errors	75
Figure 63: Bias and Integral of Position: No Errors	76
Figure 65: Position and Velocity Output: Error Condition 1, starting at $t = 1s$	76
Figure 66: Position and Velocity Output, Zoomed View: Error Condition 1, starting at $t = 1s$	77
Figure 67: Bias and Integral of Position: Error Condition 1, Starting at $t = 1s$	77
Figure 68: Bias and Integral of Position, Zoomed View: Error Condition 1, Starting at $t = 1s$	78
Figure 69: Position and Velocity Output: Error Condition 2, starting at $t = 1s$	78
Figure 70: Position and Velocity Output, Zoomed View: Error Condition 2, starting at $t = 1s$	79
Figure 71: Bias and Integral of Position: Error Condition 2, Starting at $t = 1s$	79
Figure 72: Bias and Integral of Position, Zoomed View: Error Condition 2, Starting at $t = 1s$	80
Figure 73: Position and Velocity Output: Error Condition 3, starting at $t = 1s$	80
Figure 74: Position and Velocity Output, Zoomed View: Error Condition 3, starting at $t = 1s$	81
Figure 75: Bias and Integral of Position: Error Condition 3, Starting at $t = 1s$	81

Figure 76: Bias and Integral of Position, Zoomed View: Error Condition 3, Starting at $t = 1$ s	82
Figure 77: Position and Velocity Output: Error Condition 4, starting at $t = 1$ s	82
Figure 78: Position and Velocity Output, Zoomed View: Error Condition 4, starting at $t = 1$ s	83
Figure 79: Bias and Integral of Position: Error Condition 4, Starting at $t = 1$ s	83
Figure 80: Bias and Integral of Position, Zoomed View: Error Condition 4, Starting at $t = 1$ s	84
Figure 81: Position and Velocity Output: Error Condition 5, starting at $t = 1$ s	84
Figure 82: Position and Velocity Output, Zoomed View: Error Condition 5, starting at $t = 1$ s	85
Figure 83: Bias and Integral of Position: Error Condition 5, Starting at $t = 1$ s	85
Figure 84: Bias and Integral of Position, Zoomed View: Error Condition 5, Starting at $t = 1$ s	86
Figure 85: Position and Velocity Output: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error	86
Figure 86: Position and Velocity Output, Zoomed View: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error	87
Figure 87: Bias and Integral of Position: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error	87
Figure 88: Bias and Integral of Position, Zoomed View: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error	88
Figure 89: Final Simulink Model For $y, \dot{y}, \theta_x$, and $\dot{\theta}_x$ only	89
Figure 90: Bias Calculation Filter Output for System Operating with No Position Sensor Errors	89
Figure 91: Position and Velocity Output for System Operating with No Position Sensor Errors	90
Figure 92: Bias Calculation Filter Output for System Operating with No Position Sensor Errors (Noise Only)	90
Figure 93: Control Effort for Error Condition 1, starting at $t=1$ s	91
Figure 94: Position and Velocity Output for Error Condition 1, starting at $t=1$ s	91
Figure 95: Zoomed view of Position and Velocity Output for Error Condition 1, starting at $t=1$ s	91
Figure 96: Bias Calculation Filter Output for Error Condition 1, starting at $t=1$ s	92
Figure 97: Zoomed View of Bias Calculation Filter Output for Error Condition 1, starting at $t=1$ s	92
Figure 98: Control Effort for Error Condition 2, starting at $t=1$ s	92
Figure 99: Position and Velocity Output for Error Condition 2, starting at $t=1$ s	93
Figure 100: Zoomed View of Position and Velocity Output for Error Condition 2, starting at $t=1$ s	93
Figure 101: Bias Calculation Filter Output for Error Condition 2, starting at $t=1$ s	94
Figure 102: Zoomed View of Bias Calculation Filter Output for Error Condition 2, starting at $t=1$ s	94
Figure 103: Control Effort for Error Condition 3, starting at $t=1$ s	94
Figure 104: Zoomed View of Control Effort for Error Condition 3, starting at $t=1$ s	95
Figure 105: Position and Velocity Output for Error Condition 3, starting at $t=1$ s	95

Figure 106: Zoomed View of Position and Velocity Output for Error Condition 3, starting at $t=1s$	95
Figure 107: Bias Calculation Filter Output for Error Condition 3, starting at $t=1s$	96
Figure 108: Zoomed View of Bias Calculation Filter Output for Error Condition 3, starting at $t=1s$, Zoomed	96
Figure 109: Control Effort for Error Condition 4, starting at $t=1s$	96
Figure 110: Zoomed View of Control Effort for Error Condition 4, starting at $t=1s$	97
Figure 111: Position and Velocity Output for Error Condition 4, starting at $t=1s$	97
Figure 112: Zoomed View of Position and Velocity Output for Error Condition 4, starting at $t=1s$	97
Figure 113: Bias Calculation Filter Output for Error Condition 4, starting at $t=1s$	98
Figure 114: Error Condition 17, starting at $t=1s$	98
Figure 115: Position and Velocity Output for Error Condition 17, starting at $t=1s$	98
Figure 116: Bias Calculation Filter Output for Error Condition 17, starting at $t=1s$	99

Variables

x - System States

$\dot{x}$ - Time Derivative of x

$\hat{x}$ - Estimate of System States

$\dot{\hat{x}}$ - Estimated Time-Rate of Change of System States

$\tilde{x}$ - Measured Values for System States

y - System Input

u - System Output

z - System States (alternate notation). Units are meters for position values, meters/second for velocity values, radians for angles, and radians/second for radial velocity.

A - State Matrix

B - Input Matrix

C - Output Matrix

D - Feed Forward Matrix

K - Kalman filter gain

Abbreviations

AMB – Active Magnetic Bearing (Actively controlled magnetically levitated bearing)

MIMO – Multi-Input-Multi-Output

PID – Proportional-Integral-Derivative Controller; A controller using proportional gains, integral gains, and derivative gains to control system output

SISO – Single-Input-Single-Output

VAD – Ventricular Assist Device; LVAD is for the left ventricle and RVAD is for the right ventricle

HESA – Hall Effect Sensor Array. Each array contains four sensors: positive and negative x and positive and negative y

MMAE – Multiple Model Adaptive Estimation

1.0 Introduction

Remote health monitoring of systems is an area which is increasingly studied as companies look for new ways to reduce maintenance costs and loss of income due to system failures. The ability to detect and diagnose faults also allows for better control of systems where precise performance is required [1], and detecting faults prior to system failure allows maintenance to be performed, preventing catastrophic failure. Fault tolerant operation, often accomplished using adaptive control, is a similar concept; changes in the system are compensated for such that there is little or no negative impact on system performance [2]. Fault tolerance and adaptive control may also be combined to create a system which adapts to changes while also indicating any system degradation [1,2,3].

Highly unstable systems provide a particular challenge as the operation must be tightly controlled to maintain stability. The heart pump which is being developed by the RIT heart pump project is an example of a highly unstable system where proper operation is critical. With the exception of a few specific cases, magnetically levitated bearings are highly unstable systems.

1.1 Heart Pumps/RIT Heart Pump Project

The ability to transplant human hearts from a donor to another human has significantly improved the prognosis for patients with various heart defects or diseases; heart problems which are otherwise fatal or confine the patient to a hospital bed can instead be cured with the implantation of a healthy heart from a deceased donor. However, insufficient numbers of hearts suitable for heart transplant patients necessitates the use of manufacturable devices. Artificial implantable heart pumps, also called ventricular assist devices (VADs), were developed to help fill this need while avoiding some of the problems associated with artificial hearts [4]. Ventricular assist devices aid a weak or damaged heart in pumping blood; these implantable pumps are small compared to other available devices that accomplish this same task and are portable so patients can maintain a relatively good quality of life while awaiting transplant. Figure 1 illustrates implementation of a typical LVAD device.

Figure 1: Implanted LVAD [4]

VADS were initially intended for use as a stop-gap measure until a suitable donor heart became available, however, they are now used as a permanent implant in some cases. Because of this, more recent research has focused on use as a permanent solution [4]. Ideally a heart pump would allow a patient to continue with their typical everyday activities, including various levels of physical activity and at various physical orientations (i.e. laying down, sleeping, or standing up, climbing stairs). There are currently several types of implantable heart pumps on the market, but improvements are continually sought. A project currently underway at RIT is working on creating one such improved device, with a longer service life for permanent implantation [4]. The heart pump project at RIT is working to create an improved left ventricular assist device using a magnetically levitated pump; it is hoped that the lack of contacting moving parts will provide a more durable system as well as result in less damage to the blood. The goal is to increase the service life of the pump to ten years, from the five years expected from the pumps that are currently available [4], reducing the number and frequency of surgeries for patients with permanent implants. Like the products currently on the market, the design consists of both internal components (the pump and sensors) and external components (controller and batteries).

1.2 Assessment of Need

Since patients with an implanted heart pump rely on the device for sufficient blood circulation, proper functionality is critical. In some patients, failure of a heart pump could result in death. Thus, the ability of the system to compensate for degraded component performance while also alerting the user that the system is not operating in an optimal manner could be of significant benefit to the project, and subsequently any patients who use the device.

Position sensor failure was selected as the focus for this project because position sensing error is one of the most common types of faults to occur in actively controlled magnetic bearing systems [5,6], and position sensor faults are the most common type of hardware problem experienced by the heart pump project. The goal of this project is to create a system for detecting and compensating for position sensor faults, maintaining adequate functionality despite any faults until the device can be changed. Because one of the goals of the heart pump project is to minimize the overall size of the device, especially the implantable portion, and, in addition, increasing the number of wires protruding from the patient increases the risk of infection, the fault detection and tolerance system must use only the existing sensors. These limitations exclude many otherwise feasible solutions for fault detection and fault-tolerance, so a solution which does not require sensor redundancy is required. A Kalman filter-based estimator which does not require component redundancy can be implemented, however, given the drawbacks of model-based solutions, it is unknown whether such a system is feasible for use with a highly unstable system.

1.3 History

Fault detection and fault-tolerance are well-studied fields with a wide range of solution methods. Techniques vary from simple component redundancy to adaptive, neural network-based solutions. Most research, however, considers fault detection and fault-tolerance to be distinct, unrelated areas. The thought is that fault detection is not necessary for a fault-tolerant system, and vice-versa. Prior research fault detection and fault-tolerance has also been completed for the specific case of magnetically levitated bearings.

1.3.1 Fault Detection and Fault-Tolerance

Cole et. Al [6] developed a comprehensive list of fault modes for a magnetically levitated bearing/rotor system, as shown in table 1. According to Cole et. Al [6] and Na et. Al [5] actuators/amplifier and position sensor problems are the most common fault and failure modes for a magnetic bearing system. Both faults result in unpredictable system behavior, but compensative action may be possible, and both failure modes require component redundancy for fault tolerance, according to the authors.

Table 1: Fault Modes for an Active Magnetic Bearing System [6]

Fault Mode	Internal / External	Effect Without Tolerant Control	Fault Tolerance Feasibility
Power electronics or amplifier faults	Internal	Unpredictable	Compensative action may be possible.
Power electronics or amplifier failure	Internal	Stability loss	Requires component redundancy
Transducer faults	Internal	Unpredictable	Compensative action may be possible.
Transducer complete failure	Internal	Stability loss	Requires component redundancy
Loss of I/O board channel	Internal	Stability loss	Requires component redundancy
Bearing magnet coil failures	Internal	Stability loss	Requires component redundancy
Computer hardware failures	Internal	Complete loss of control likely	Requires component redundancy
Computer software errors	Internal	Complete loss of control likely	Requires fail-safe programming
Rotor impact (not rotor-stator)	External	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Rotor mass loss	External	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Base motion	External	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Rotor deformation	External/ Internal	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Sudden changes in loading	External	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Rotor rub	External/ Internal	Abnormal rotor vibration	Requires suitable controller design and sufficient control force
Cracked rotor	External/ Internal	Abnormal rotor vibration	Requires suitable controller design and sufficient control force

Various methods for implementing fault-tolerance and fault detection in AMB systems have been investigated by previous authors. Herzog et. Al [7] used generalized notch filters for unbalance compensation in magnetic bearings with multi-variable feedback. Because

rotating machinery systems are incredibly difficult and expensive to balance perfectly, it is easier and less expensive to filter out synchronous vibrations, preventing the negative effects instead of entirely preventing vibrations. Unlike observer-based methods, generalized notch filters do not require an accurate plant model to work effectively and they provide closed-loop system stability and good transient behavior. While generalized notch filters work well for periodic errors such as rotational unbalance, they are not adequate for sudden failures or the type of time-varying drift which might be exhibited by a position sensor.

Next, Schroder et. Al [8] looked at fault-tolerance for amplifier and coil failures for a model of a Rolls-Royce turbo machine rotor supported by active magnetic bearings. A centralized control reconfiguration algorithm was compared to decentralized algorithms and different bearing configurations were used; centralized control with geometric fault compensation was found to be better than decentralized control in all cases, and increased numbers of bearing poles result in greater fault tolerance. Demetriou et. Al [9] worked to create a method for detection and possible diagnosis of faults using only one observer; the adaptive observer first serves as a fault detection observer, and then once a fault is detected, as a diagnostic observer. A fault detection observer monitors the signal and detects when the values are not as expected, and a diagnostic observer diagnoses the error based on the faulty signal. The scheme requires the plant model and bounds for modeling uncertainty to be known, but provided good results when tested by simulation.

In 2000, Cole et. Al [10] attempted to increase viability of rotor/magnetic bearing systems for industrial purposes by working on improved, more fault tolerant control systems. They used a reconfigurable control scheme with built-in fault detection; neural networks trained to detect faults combined with a multi-variable H_∞ controller which is reconfigured to maintain stability when a fault is detected. An H_∞ controller is a system of controllers plus a method for selecting the optimal one based on the current operating conditions; when the operating conditions are changing, the active controller can be changed during operation to a different controller which is better suited to the current conditions. Their control scheme is applicable to many types of system configurations, but can only detect specific faults at specific operating speeds, so the greater the range of operating conditions or system complexity, the

greater the number of necessary fault detectors. Na et. Al [5] also aimed to increase the reliability of magnetic bearings in order to increase the possibilities for industrial applications. They used redundant coils with current grouping to avoid increasing the number of controller outputs. In the event of a coil failure, the affected coil pair is shut down completely and the coil currents are redistributed according to an adaptive distribution matrix; distribution matrices are calculated using a Lagrange Multiplier optimization method. This method works for up to five failed coils, depending on the coil locations, but requires complete component redundancy, and the method used for detecting the fault was not covered.

Sahinkaya et. Al [11] did additional work on synchronous vibration control in 2001, using an open-loop adaptive control scheme for fault detection and tolerance. Data is stored during nominal operation, and then operating data is compared to this stored data; significant deviation indicates the presence of a fault. Steps taken to mitigate effects of a fault depend on the fault detected, and the controller can be re-optimized if the fault cannot be identified. Any controller updates are stored in the system so that they can be analyzed later to detect slow changes in the system. Their controller was tested experimentally and provided good results.

Montie and Maslen [12] took a slightly different approach to fault-tolerance, aiming to reduce the number of possible failure modes in a system, in addition to developing fault-tolerance capabilities, by using self-sensing instead of the typical position sensors. The reasoning for this is that adding position sensors not only adds cost, but also increases the number of components which might fail. Time-rate of change of the coil currents is used to estimate the rotor position, and a parameter estimator is run simultaneously with the system. The set-up was tested using a simulation and provided good results.

Cole et. Al [6] studied and classified possible failure modes for a magnetic bearing system, then implemented methods for increasing tolerance to these faults. Their interest was in allowing systems to shut down properly and safely for repairs in the event of a component failure, not to allow the system to continue operating normally.

Zhang et. Al [2] worked on an adaptive fault-tolerant controller for non-linear uncertain systems. This system is not only fault tolerant, but also detects and diagnoses faults, and is not designed specifically with AMBs in mind. Because prior research had focused primarily on fault detection alone, or fault tolerance alone, they felt the links between the two fields had not been adequately established. Fault detection is typically not considered necessary for a fault tolerant system because a system with good fault tolerance should run normally in the presence of a fault, however, there are still many instances in which it would be useful to know that a system component is not operating as specified. Their nominal controller is sufficiently robust to keep signals bounded when a fault occurs, then once the fault has been detected, the controller is updated to better control operation. Once the fault has been isolated, the controller is updated again based on information from the fault diagnosis system. Faults can only be isolated if they have been anticipated to occur in the system, so if a fault which cannot be identified occurs, the controller is only updated once. The system was tested via simulation and provided good results.

Noh et. Al [13] considered the major barrier for widespread use of AMBs in industrial applications to be the different failure modes compared to conventional bearings. They designed and implemented a fault-tolerant magnetic bearing system for a turbo-molecular vacuum pump which was tolerant to the two major fault modes for magnetic bearings: faults in position sensors and actuator/amplifier faults. They used bias linearization in combination with linear power amplifiers for actuator/amplifier faults and redundant position sensors for position sensing.

Unlike Zhang et. Al [2], Garimella and Yao [1] considered fault-tolerance to include fault detection and isolation. Like [2], they also worked on an adaptive fault-tolerant controller which allows for modeling errors. Model-based fault detection systems are widely used because they are easy to use and respond rapidly to abrupt failures, however, these systems assume that a perfect mathematical model for the controlled system is available. They used an adaptive, non-linear, online detection system which works by monitoring residuals (differences between measured signals and desired or estimated signals) and can differentiate

between modeling errors and faults. Chowdhury [3] developed a fault detection system for closed-loop systems which is especially helpful for adaptive systems. This system makes use of both output residuals and controller residuals combined into a specific function for indicating faults.

Xue and Jiang [14] combined aspects of both neural networks and fuzzy logic in order to make use of the distinct advantages of each while minimizing the drawbacks. Fuzzy logic is good for integrating expert (human) knowledge into fault detection, however, if there are more than a few rules the system become extremely complex. Meanwhile, neural networks are self-learning, simulate and model nonlinear objects well, and can handle noise and corrupted data very well, but cannot make use of expert knowledge. Simulations showed their ‘fuzzy neural network system’ to be able to track and compensate for an unknown fault, however, an accurate system model is required, and any changes to that model require neural network retraining.

Paek et. Al [15] looked at redundancy as a relatively simple method for fault detection in AMBs in a rotating machinery system. A sensor with 16 coils was rewound to provide fewer, redundant signals, with less sensor resolution. Their system was tested experimentally on a magnetically levitated turbo-molecular vacuum pump and provided good results.

Some research has focused specifically on methods which are applicable to unstable systems. Most solutions make use of residuals, like the method used by [2] and controllers or systems of controllers which can maintain stability in the presence of a fault. Like most of the other fault detection or fault-tolerance methods, redundancy is common requirement for many of the solutions, and thus they cannot be directly applied to this problem.

According Kinnaert et. Al [16], most fault-detection schemes do not consider plant stability a criteria for successful implementation of the scheme. This is due to the assumption of a perfect plant model. They showed that unless the controller is able to provide stability, residual methods are unsuitable for unstable systems because of the differences between the actual plant and the estimated plant model. They also showed how to quantify the effect of

modeling uncertainties on the residual generator. Varga, [17] and [18] addresses the development of a general reliable computational approach to creating fault detection filters which are suitable for use on all types of systems, including unstable systems. An H_∞ or H_2 optimization approach is used to produce filters for residual generation. These residuals would then be used in a system such as [2] or [16] which uses residuals to detect and accommodate faults.

Cardoso and Dourado [19] used a model-based strategy with an H_∞ controller, which they applied to an unstable, nonlinear process. Residual signals were used by a supervisory system to detect and identify faults, and then the H_∞ controller allowed for the control to be adjusted based on the fault. As with similar work, only anticipated fault conditions resulting in the expected output can be properly diagnosed and accounted for, and sensor redundancy is required for fault-tolerance. Zhang et. Al [20] designed a linear quadratic state feedback regulator for open-loop unstable systems with actuator redundancy. Their design was validated using a maglev train model. The advantage of this method is that it is robust to various system failures without requiring fault detection or diagnosis, however, component redundancy is necessary.

Few of these fault-detection and fault-tolerance schemes have actually been tested on real systems. Most of the schemes were intended for large-scale industrial equipment, and none were developed specifically for a heart pump application. Furthermore, most of the research considered either fault detection or fault-tolerance separately.

1.3.2 Heart Pumps and Magnetically Levitated Bearings

A major difficulty with using mechanical bearings in artificial hearts is the formation of blood clots in the bearings and shaft [21]. Damage to the blood constituents and insufficient flow through areas of the bearing result in blood clot formation; these clots eventually break loose and continue through the circulatory system, potentially causing serious complications. Sealing problems and system volume are other drawbacks of standard mechanical bearings according to [21]. Shen et. Al [21] investigated the use of magnetically levitated bearings in an artificial heart in 1999 in order to alleviate these problems, but did not test their proposed

system experimentally. A follow-up study in 2000 included experimental testing, but noise in the system prevented the required rotational speed from being achieved [22]. In 2005, Okada et. Al [23] designed and implemented a heart pump system using magnetically levitated bearings, but their results were disappointing.

Magnetically levitated bearings consist of an outer set of magnets which are used to levitate a magnetic rotor. They have no contacting moving parts so there is no friction in the bearing; as a result, there is no wear due to friction and there are no frictional losses. Because of these advantages, magnetic bearings have great potential for use in industrial and other applications. However, control and reliability difficulties are currently limiting the number of feasible applications. Previous research has looked at various methods for overcoming these shortcomings, and several potential solutions have been developed. A selection of this research is presented in the following sections.

1.3.3 Control of Magnetically Levitated Bearings

Typical magnetic bearings consist of sets of opposing electromagnets which are used to suspend a magnetic rotor. Magnetically levitated bearings generally require active control in order to operate stably; for active bearing control the strength of the electromagnetic field for each magnet is varied to keep the rotor centered and so rotor position feedback is necessary. This feedback is typically accomplished using external position sensors (optical or proximity). Work on self-sensing has also been completed; current through the electromagnet coils can theoretically provide rotor position information, however, successful experimental results have not been obtained. In some bearings, both sets of magnets are permanent magnets. In the case where only permanent magnets are used, the bearing is a passively controlled bearing, which means there is no means of directly controlling the bearing; a passively controlled bearing typically must be used in conjunction with actively controlled bearings to provide stability.

PID control is relatively common for control of AMBs; it is used directly in the work of Barthod and Lemarquad [24] and Zhou and Twardowski [25]. Barthod and Lemarquad's [24] goal was to design and implement a levitation system for magnetic bearings suitable for

use in high speed rotation applications. Four axes were actively controlled and the fifth was controlled passively, where the five axis are x , y , z , and two rotational. Zhou and Twardowski [25] worked towards implementing magnetically levitated bearings to drive pumps in spent nuclear fuel reprocessing facilities; the use of magnetically levitated bearings would reduce the maintenance requirements of the pumps, and consequently reduce the need to breach the biological radiation shield, which increases exposure risks for workers. The effects of varying the P and D gains were investigated, and the physical system and controller were tested experimentally on a pump.

More recent work has also made use of PID control. Fangmin and Xu [26] used a combination of PI and PID control in their design of a fully digital AMB control and amplifier system. They considered the problem as a five degree of freedom control problem, where the behavior of each degree of freedom is considered to be independent from the others; five eddy current sensors were used to determine spindle position and five separate control chips controlled the position for each degree of freedom. The system was tested experimentally but the results were somewhat unclear. Okada et al. [23] used a method similar to standard PID control for a small artificial heart pump with a magnetically levitated rotor. A double mixed flow pump was driven by a bidirectional axial-type self-bearing motor using permanent magnets. Two design iterations were completed, however, the pump still did not meet specifications when tested on cow blood; while the pump operated sufficiently well when pumping air and water, it was not able to achieve the necessary pressure and flow rates when pumping blood. In addition, insufficient damping in the system resulted in blood constituent damage.

Various forms of adaptive control have been used as well, including a Lyapunov-based adaptive control law by Green and Craig [27]. They consider AMBs to be a potential solution for the machine-tool industry's need for high-speed precision machining, provided that the bearings can be sufficiently well controlled. They selected adaptive control because it does not require an accurate system model and because it lacks some of the drawbacks of previously tried non-linear control designs, such as sliding mode control and time-delayed control. They considered the more general case of magnetic levitation, which, with certain

conditions applies to AMBs. Like [26] they assumed the dynamics of each axis were uncoupled, allowing them to apply one SISO control law five times to control an AMB; this simplification is necessary to make adaptive control feasible to use. Their work built on previous research on Lyapunov-based adaptive and backstepping controllers.

Sliding mode control has also been used with apparent success by Yeh and Chung [28], among others, however, according to [27] there are some serious drawbacks to the use of sliding mode control on a magnetic bearing. Large control efforts are required and chattering occurs in the control response; the chattering can be reduced by modifying the discontinuity, but the asymptotic stability of the error dynamics is forfeited. Yeh and Chung [28] used sliding mode control with two separate controller components: the first is a nominal control part to linearize the nonlinear dynamics (feedback linearization) and the second part compensates for unknown parameters to provide robust control. The controller was tested on a thrust bearing of a magnetically levitated rotor, and the test results showed that the control system can maintain stability and consistent performance.

Another type of controller which has been used is the H_∞ controller, including by Cole et. Al [10]. They successfully used neural networks that were trained to detect faults in conjunction with a multivariable H_∞ controller which was reconfigured to maintain stability when a fault was detected.

1.4 Scope

The goal of this project was to assess the feasibility of a Kalman filter based fault detection and fault tolerance method for a highly unstable system, specifically the RIT heart pump. The system included combined fault detection and fault-tolerance and was to work within the limitations specific to the heart pump system. In addition, experimental work was done to determine the effects of position sensor fault conditions on the system, and that information was used in creating an accurate simulation of the system and then a method of detecting and compensating for those faults.

Faults were simulated using the currently available Simulink model of the system and experimental results were obtained for selected fault conditions to validate the results obtained from the model. Then an estimator for position sensor output was developed. Testing was completed using MATLAB/Simulink, and then a sensitivity analysis was completed to determine the sensitivity of the estimator to plant model inaccuracies. Final testing was completed using the heart pump test stand to validate the results of the sensitivity analysis.

2.0 Heart Pump

The primary goals of the RIT heart pump project are to create a heart pump with a longer service life that is less damaging to the blood and is smaller than the products currently on the market. Magnetically levitated bearings were selected to reduce mechanical wear and stress on the blood.

2.1 Hardware and Configuration

The heart pump consists of an outer housing containing a rotor, two active magnetic bearings to levitate the rotor, and two Hall effect sensor arrays (HESAs) to measure the position of the rotor. The heart pump assembly diagram is shown in Figure 2, a picture of one of the test assemblies is shown in Figure 3, and Figure 6 is a picture of a HESA sensor.

Figure 2: Heart Pump Assembly View [4]

A brushless DC motor drives the rotor, and the spinning motion of the rotor forces the blood through the pump. The rotor, shown in Figure 5, is hollow, and the permanent magnets are placed inside. The pump, the brushless DC motor which drives the pump, and sensors required for control of the pump are implanted in the patient, and the power and control system is connected by wires protruding from the patient.

Figure 3: Complete LVAD System [4]

Figure 4: Rotor Magnet Polarity and Arrangement [4]

Figure 5: Impeller [4]

Figure 6: Sensor Arrangement, End View of a HESA sensor installed in the heart pump

Each HESA sensor provides four position measurements, which are subtracted to obtain the two measurements which are used in the controller, as shown in Figure 7.

Figure 7: Diagram of signal path from the HESA sensors to the pump

2.2 Controller

Currently, a PID controller is used for rotor levitation. PID control stands for Proportional-Integral-Derivative control; a different gain value is determined for each input signal, the integral of the input signal, and the derivative of the input signal. During operation, the gains are applied to the appropriate signals and added together to get the adjusted signal, which is used as input to the plant model; Figure 8, below, shows a general PID controller.

Figure 8: PID Controller

Both the controller and dynamic model were developed as part of the heart pump project and provided for use in this project. The dynamic model and plant model derivation are shown in Appendix D. The basic plant model and controller are shown below in Figure 9.

2.3 System Model

The system model consists of the plant model and controller. The plant model is a representation of the dynamic behavior of the heart pump, and the controller adjusts the input to the plant (i.e. magnetic field strength) in order to achieve the desired plant output. The controller and equations for modeling the heart pump system were developed by the heart pump project and used to construct a model of the system using Simulink. The original controller and Simulink model were provided by Dr. A. Crassidis via personal communication. This Simulink model was used as the starting point for this thesis.

Figure 9: Basic Simulink Model. ‘u1’-‘u4’ are controller output values. The dead zone block limits the controller force to better mimic the actual system limitations.

The continuous-time plant model is:

$$\begin{aligned} \dot{z} &= Az + Bu \\ y &= Cz + Du \end{aligned} \quad (1)$$

where:

u = system input (voltage to electromagnets)

z = system states

y = system output (rotor position)

A = state matrix

B = input matrix

C = output matrix

D = feed forward matrix

and

$$z = \begin{bmatrix} x \\ \dot{x} \\ \theta_y \\ \dot{\theta}_y \\ y \\ \dot{y} \\ \theta_x \\ \dot{\theta}_x \end{bmatrix}; \quad y = \begin{bmatrix} x \\ \theta_y \\ y \\ \theta_x \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-(k_1+k_2+k_3)}{m} & 0 & \frac{(k_1d_1-k_2d_2-k_3d_3)}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{(k_1d_1-k_2d_2-k_3d_3)}{J} & 0 & \frac{-(k_1d_1^2+k_2d_2^2+k_3d_3^2)}{J} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-(k_1+k_2+k_3)}{m} & 0 & \frac{(k_1d_1-k_2d_2+k_3d_3)}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{(k_1d_1-k_2d_2+k_3d_3)}{J} & 0 & \frac{-(k_1d_1^2+k_2d_2^2+k_3d_3^2)}{J} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{m} & \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{-d_1}{J} & \frac{d_2}{J} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & \frac{1}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-d_1}{J} & \frac{d_2}{J} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 10: Illustration of rotor with coordinate system

This plant is for a continuous-time system, and since the actual system is sampled and controlled in discrete-time, the model was discretized for this project using the MATLAB built in continuous to discrete function.

The discrete-time form of the plant model is denoted by:

Discrete Plant Model:

$$\begin{aligned} z_{k+1} &= \Phi z_k + \Gamma u_k \\ y_k &= C z_k + D u_k \end{aligned} \quad (2)$$

where

Φ = discrete state matrix

Γ = discrete input matrix

and C and D are the same as for the continuous form above (the output and feed forward matrices are the same for the continuous and discrete forms). C can be alternately noted as H for the discrete form.

3.0 Fault Detection and Fault Tolerance Method Selection

3.1 Needs

In order to be useful to the heart pump project, the fault detection and fault-tolerance system must use only the existing components and require no additional wires between the implanted and external portion of the heart pump. The system must also not interfere with normal functionality or reduce the speed of operation. In addition, filtering of the system output is desired for normal operation because there is a significant noise component in the position data. Noise filtering is of more immediate use for the project because current operational difficulties could be a result of measurement noise. While fault detection could ultimately be a very useful addition to the heart pump project, proper performance with functional system components is the first priority.

3.2 Selection

Because there are eight position sensors for determining four position values, there is built-in redundancy in the system. However, since only the differenced values are used for control, this redundancy is not utilized. Final system configuration has not yet been determined, but the differencing will likely take place in the implanted portion, thus making use of the sensor redundancy difficult or impossible. Sensor redundancy would increase the number of available options for fault detection and fault tolerance.

Self-sensing is another possibility for adding redundancy; however this method has not yet been fully developed or proven to work. Thus it was considered to be too much of a risk for this project. Since the scope of this project was narrowed to include only position sensor faults, many of the methods from the literature were unnecessarily complex.

Sensor Bias Estimation using a Kalman filter was selected as the final choice because it is a relatively straightforward method of adding fault-tolerance and it provides data filtering as well. In addition, it is independent of the controller and can be easily updated as the plant model is revised. One drawback is that it is a model-based method and thus depends on a relatively accurate plant model to provide good results.

3.3 Kalman Filter

Kalman filters are a type of sequential state estimator. A sequential state estimator estimates the next state based on the current state and measurement information, and the plant model. The model output and the measurement data are weighted based on their reliability and combined to produce estimated actual state values; thus an estimator also functions as a filter, so the two terms can be used interchangeably. Design of filters requires selection of system poles to achieve a desired characteristic equation [29]; for most types of filters the poles are selected on a case by case basis and adjusted until the correct behavior is achieved. The Kalman filter provides a theoretical basis for pole selection based on a random process for the measurement and model error; exact error information is not needed, but certain assumptions are made about the nature of the error, such as that it is a zero-mean Gaussian process with a known variance. A Kalman filter requires only the current measurement data and estimated position to calculate the next current estimated position.

There are several different Kalman filter variations to cover different types of systems and different types of noise. A standard steady-state discrete-time Kalman filter was used for the initial steps of this work. The process and measurement noise are assumed to be random, uncorrelated, zero-mean Gaussian processes. The assumption of random noise means that the error for the current time is not influenced by or related to the error at any other point in time, and the assumption of uncorrelated noise means that the process noise is unrelated to the measurement noise. The general Kalman filter is defined below [29,30].

General Kalman Filter:

model:

$$\hat{x}_{k+1} = \Phi_k x_k + \Gamma_k u_k + Y_k w_k \quad (3)$$

$$\tilde{y}_k = H_k x_k + D_k u_k + v_k \quad (4)$$

where w_k is the process noise and v_k is the measurement noise

$$w_k \sim N(0, Q_k)$$

$$v_k \sim N(0, R_k)$$

and Y_k is the state correction matrix

gain:

$$K_k = \frac{P_k H_k^T}{H_k P_k H_k^T + R_k} \quad (5)$$

update:

$$\hat{x}_k = \hat{x}_k^- + \Phi_k K_k [\tilde{y}_k - H_k \hat{x}_k - D_k u_k] \quad (6)$$

$$P_k = [I - K_k H_k] P_k^- \quad (7)$$

propagate:

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k + \Gamma_k u_k \quad (8)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Y_k Q_k Y_k^T \quad (9)$$

The model (equations (3) and (4)) is the original system model (equation (2)) with noise added. The measured values at the current time step k , $\tilde{y}_k$, are assumed to be the actual value, $y_k = H_k x_k + D_k u_k$, plus some unknown measurement noise v_k which follows a zero-mean Gaussian distribution with a known variance. The system states for time-step $k+1$ are assumed to be the value calculated according to the known model, $\hat{x}_{k+1}^- = \Phi_k x_k + \Gamma_k u_k$, plus unknown process noise $Y_k \cdot w_k$. The process noise can also be thought of as model uncertainty. Like v_k , w_k is assumed to follow a zero-mean Gaussian distribution with a known variance, and Y_k is a weighting matrix, allowing the noise value to be applied differently to different states.

Preliminary values for P and x are used to compute initial Kalman Filter gains, and the filter is used to compute an estimate for the current time step. Then the error covariance P is calculated for the current time-step, the propagation equations are used to compute estimated P and x values for the next time-step, $k+1$. Then the process repeats continuously: gain values (equation 5) are calculated for the current time-step, and that value is used to update the state estimate and covariance values (equations 6 and 7, respectively). Then the states and covariance values are estimated for the next time step (equations 8 and 9). Figure 11 illustrates this process.

Figure 11: Illustration of Kalman Filter Functionality [30]

If the variance of the process noise is high compared to the variance of the measurement noise, then the measurements are considered to be more reliable than the model, and the resulting Kalman filter gains are small. The effect of small filter gains is that the measured values are weighted more highly than the model output in determining the estimated state. If the variance of the measurement noise is high compared to the process noise, then the Kalman gains are larger, and the model output is weighted more highly than the measured values.

In the recursive form of the Kalman Filter [29], equations (7) and (9) are combined, and equations (6) and (8) are combined, reducing the number of required computations.

General Kalman Filter (Recursive Form):

model:

$$\hat{x}_{k+1} = \Phi_k x_k + \Gamma_k u_k + Y_k w_k \quad (10)$$

$$\tilde{y}_k = H_k x_k + v_k \quad (11)$$

estimate :

$$\hat{x}_{k+1} = \Phi_k \hat{x}_k + \Gamma_k u_k + \Phi_k K_k [\tilde{y}_k - H_k \hat{x}_k] \quad (12)$$

gain :

$$K_k = \frac{P_k H_k^T}{H_k P_k H_k^T + R_k} \quad (13)$$

covariance:

$$P_{k+1} = \Phi_k P_k \Phi_k^T - \Phi_k K_k H_k P_k \Phi_k^T + Y_k Q_k Y_k^T \quad (14)$$

In the steady-state case, equations (13) and (14) quickly converge to constant values, so only the estimate is calculated at every time-step. Since the covariance value is constant, so the filter gain is also constant and they do not need to be continually recalculated.

Steady-State Kalman Filter [29]:

model:

$$\hat{x}_{k+1} = \Phi x_k + \Gamma u_k + Y w_k \quad (15)$$

$$\tilde{y}_k = H x_k + v_k \quad (16)$$

estimate :

$$\hat{x}_{k+1} = \Phi \hat{x}_k + \Gamma u_k + \Phi K [\tilde{y}_k - H \hat{x}_k] \quad (17)$$

gain :

$$K = \frac{P H^T}{H P H^T + R} \quad (18)$$

covariance:

$$P = \Phi P \Phi^T - \Phi K H P \Phi^T + Y Q Y^T \quad (19)$$

In the case of the heart pump, the random noise covariance should be constant when the pump is running at steady-state, so the steady-state Kalman filter can be used to filter out the random noise. MATLAB's built-in Kalman function creates a steady-state Kalman Filter.

3.4 Multiple Model Adaptive Estimation

The Kalman filter assumes that both the process and measurement noise variances are known. In actual systems, this is usually not a good assumption [30]. One way to deal with this problem is by using multiple model adaptive estimation (MMAE). Instead of one Kalman filter, MMAE uses a bank of Kalman filters; each filter has a different value for the unknown or variable parameter. The probability that a specific filter is correct is calculated, and that probability is used to weight the output from that filter relative the output from other filters. This adaptive method was investigated as an option for optimizing the measurement noise variance value as errors occur in the system. For this application, the measurement

noise variance R_k was the unknown parameter, but the modeling error, Q_k , is more commonly the unknown value.

The probability that a given R value is correct, given the $\tilde{y}$ values measured for each time-step, including the current time k , is denoted as $p(R^{(l)} | \tilde{Y}_k)$, where $R^{(l)}$ is the measurement noise variance, R , for the specific filter l , $\tilde{Y}_k = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_k\}$. Since the measurement noise variance is constant for each filter, $R_k^{(l)} = R^{(l)}$.

Combining Bayes' Rule:

$$p(R^{(l)} | \tilde{Y}_k) = \frac{p(\tilde{Y}_k | R^{(l)})p(R^{(l)})}{p(\tilde{Y}_k)} \quad (20)$$

where:

$p(R^{(l)})$ = prior/marginal probability of $R^{(l)}$ with no knowledge of $\tilde{Y}_k$

$p(R^{(l)} | \tilde{Y}_k)$ = conditional probability of $R^{(l)}$ given $\tilde{Y}_k$

$p(\tilde{Y}_k | R^{(l)})$ = conditional probability of $\tilde{Y}_k$ given $R^{(l)}$

$p(\tilde{Y}_k)$ = prior/marginal probability of $\tilde{Y}_k$

with the total law of probability:

$$p(\tilde{Y}_k) = P(X = \tilde{Y}_k) = \sum p(\tilde{Y}_k | R^{(j)})p(R^{(j)}) \quad (21)$$

results in (combining equations (20) and (21)):

$$\therefore p(R^{(l)} | \tilde{Y}_k) = \frac{p(\tilde{Y}_k | R^{(l)})p(R^{(l)})}{\sum p(\tilde{Y}_k | R^{(j)})p(R^{(j)})} \quad (22)$$

and

$$p(\tilde{y}_k | \tilde{Y}_{k-1}, R^{(l)}) = p(\tilde{y}_k | \hat{x}_k^{-(l)}) \quad (23)$$

This means that the probability of $\tilde{y}_k$ given the previous measurements and $R^{(l)}$ is the same as the probability of $\tilde{y}_k$ given the current state estimate, $\hat{x}_k^{-(l)}$, and its associated covariance (all involved distributions are Gaussian). In other words, $\hat{x}_k^{-(l)}$ (which is the mean

of $p(\tilde{y}_k | \tilde{Y}_{k-1}, R^{(l)})$ and its associated covariance provide an equally good but less computationally intensive representation of all the data up to time k-1, as clarified by Yang Cheng via email.

The probability density function for a Normal Distribution with a mean of μ and variance σ^2 is [31]:

$$p_{\mu, \sigma^2}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \quad (24)$$

For a zero mean normal distribution:

$$p_{0, \sigma^2}(x) = \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \quad (25)$$

For a Kalman Filter:

$$x = e_k^{(l)} \quad (26)$$

$$\sigma^2 = H_k P_k^{-(l)} H_k^T + R_k \quad (27)$$

So, substituting equations (26) and (27) into equation (24):

$$p(\tilde{y}_k, Q^{(l)}) = L_0^{(l)} = \frac{\exp\left[-\frac{1}{2} e_k^{(l)T} (H_k P_k^{-(l)} H_k^T + R_k) e_k^{(l)}\right]}{\{\det[2\pi(H_k P_k^{-(l)} H_k^T + R_k)]\}^{1/2}} \quad (28)$$

and then equation (28) into equation (23):

$$p(\tilde{y}_k | \tilde{Y}_{k-1}, Q^{(l)}) = w_k^{(l)} = w_{k-1}^{(l)} L_0^{(l)} = w_{k-1}^{(l)} \frac{\exp\left[-\frac{1}{2} e_k^{(l)T} (H_k P_k^{-(l)} H_k^T + R_k) e_k^{(l)}\right]}{\{\det[2\pi(H_k P_k^{-(l)} H_k^T + R_k)]\}^{1/2}} \quad (29)$$

where

$$\begin{aligned}
e_k^{(l)} &= \tilde{y}_k - H_k \hat{x}_k^{-(l)} \\
w_k^{(l)} &= w_{k-1}^{(l)} L_0^{(l)} \\
w_k^{(l)} &\leftarrow \frac{w_k^{(l)}}{\sum_{j=1}^M w_k^{(j)}} \\
\hat{x}_k &= \sum_{j=1}^M w_k^{(j)} \hat{x}_k^{(j)}
\end{aligned}$$

3.5 Sensor Drift Rate Estimation

Certain types of sensors are known to drift over time, or provide changing measurements for the same measurement conditions. If redundant sensor information is available, with one sensor measuring a property and the other sensor measuring the rate of change of that property, it is possible to use the output of the two sensors to estimate any sensor drift which occurs in the rate sensor.

For example, measuring the angular position of an object using both an angle measurement and a rotational velocity value, where the rotational velocity value drifts randomly at a rate with known statistical bounds [29]; the measured velocity, $\tilde{\omega}$, is a function of the actual rotational speed (derivative of rotational position, θ), rate sensor drift rate, and white noise:

$$\tilde{\omega} = \dot{\theta} + \beta + \eta_v \quad (30)$$

where:

β = rate sensor drift rate

η_v = white noise, $\eta_v \sim N(0, \sigma_v^2)$

$\dot{\beta} = \eta_u$, $\eta_u \sim N(0, \sigma_u^2)$, and σ_u^2, σ_v^2 are experimentally determined variances

Estimated states are determined using the measured value and the estimated bias value:

$$\dot{\hat{\theta}} = \tilde{\omega} - \hat{\beta} \quad (31)$$

$$\dot{\hat{\beta}} = 0 \quad (32)$$

The discrete-time error propagation is given by:

$$\begin{bmatrix} \theta_{k+1} - \hat{\theta}_{k+1} \\ \beta_{k+1} - \hat{\beta}_{k+1} \end{bmatrix} = \Phi \begin{bmatrix} \theta_k - \hat{\theta}_k \\ \beta_k - \hat{\beta}_k \end{bmatrix} + \begin{bmatrix} p_k \\ q_k \end{bmatrix} \quad (33)$$

where:

$$\Phi = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \quad (34)$$

A discrete-time Kalman filter can be applied to the system to estimate the sensor bias. The measured velocity, $\tilde{\omega}$, is used as the system input, u , and the position sensor output, $\tilde{\theta}$, is used as the measurement input, $\tilde{y}$, where $\tilde{y}_k = \theta_k + \nu_k$. The two system states are attitude angle, θ , and sensor bias, β . The state-space matrices are then:

$$\Phi = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}; \quad \Gamma = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix}; \quad H = [1 \quad 0] \quad (35)$$

Substituting equation in (35) into equation (12), and setting

$$x = \begin{bmatrix} \theta_{k+1} \\ \beta_{k+1} \end{bmatrix}, \text{ and } K = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

the resulting equations are then:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \Delta t \hat{\beta}_k + \Delta t \tilde{\omega} + (k_1 - \Delta t k_2) [\tilde{y}_k - \hat{\theta}_k] \quad (36)$$

$$\hat{\beta}_{k+1} = \hat{\beta}_k + k_2 [\tilde{y}_k - \hat{\theta}_k] \quad (37)$$

So the filter output is then a position value and a bias value; the bias value can be subtracted from the measured value to provide a better estimate of the true position.

3.6 Covariance Analysis

The square root of the covariance is the standard deviation. Statistically, 68% of all data following a normal distribution should fall within one standard deviation of the expected value and 99.7% of the data should fall within three standard deviations [31]. The covariance values determined from the bias estimation filter can also be used to determine whether the error falls within expected bounds; error values which are outside of the three-sigma limits may signify a sensor error.

3.7 Design

First, a basic steady-state Kalman Filter was designed using MATLAB's built-in Kalman function, and this filter was tested using the system model with noise added. The original Simulink model was then digitized, a discrete steady-state Kalman filter was designed and tested, and the results were compared to the continuous system. Next a general Kalman filter was applied to the digitized model with steady-state conditions (which should theoretically provide the same results as the steady-state Kalman filter) and the results were compared to the steady-state filter. The Simulink model was then modified to accommodate the addition of fault conditions, and the effects of various faults on the system were observed.

Next, MMAE was added to the model, and the model was tested under various fault conditions. It was determined that using one R value for each filter, and one weighting factor for each filter output (the likelihood function returns a single value, not an array of values) provided insufficient accuracy. Since it is assumed by the state-space model equations that the degrees of freedom are independent from one another, the state-space model was split up into sub-models.

The original state-space model was:

$$z_{k+1} = \Phi z_k + \Gamma u_k \quad (38)$$

$$y_k = C z_k + D u_k \quad (39)$$

$$\text{where: } z = \begin{bmatrix} x \\ \dot{x} \\ \theta_y \\ \dot{\theta}_y \\ y \\ \dot{y} \\ \theta_x \\ \dot{\theta}_x \end{bmatrix}; \quad y = \begin{bmatrix} x \\ \theta_y \\ y \\ \theta_x \end{bmatrix}; \quad u = \begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \end{bmatrix};$$

This was changed to:

$$z_{1_{k+1}} = \Phi_1 z_{1_k} + \Gamma_1 u_{1_k} \quad (40)$$

$$y_{1_k} = C_1 z_{1_k} + D_1 u_{1_k} \quad (41)$$

$$\text{where: } z_1 = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}; \quad y_1 = [x]; \quad u_1 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix};$$

$$z_{2_{k+1}} = \Phi_2 z_{2_k} + \Gamma_2 u_{2_k} \quad (42)$$

$$y_{2_k} = C_2 z_{2_k} + D_2 u_{2_k} \quad (43)$$

$$\text{where: } z_2 = \begin{bmatrix} \theta_y \\ \dot{\theta}_y \end{bmatrix}; \quad y_2 = [\theta_y]; \quad u_2 = \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix};$$

$$z_{3_{k+1}} = \Phi_3 z_{3_k} + \Gamma_3 u_{3_k} \quad (44)$$

$$y_{3_k} = C_3 z_{3_k} + D_3 u_{3_k} \quad (45)$$

$$\text{where: } z_3 = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}; \quad y_3 = [y]; \quad u_3 = \begin{bmatrix} u_{13} \\ u_{14} \end{bmatrix};$$

$$z_{4_{k+1}} = \Phi_4 z_{4_k} + \Gamma_4 u_{4_k} \quad (46)$$

$$y_{4_k} = C_4 z_{4_k} + D_4 u_{4_k} \quad (47)$$

$$\text{where: } z_4 = \begin{bmatrix} \theta_x \\ \dot{\theta}_x \end{bmatrix}; \quad y_4 = [\theta_x]; \quad u_4 = \begin{bmatrix} u_{13} \\ u_{14} \end{bmatrix};$$

A filter gain and weighting factor were calculated for each sub-model for each R value.

While the MMAE was able to accurately calculate the variance of the signal, the filters were unable to remove the bias caused by any of the fault conditions, and instead amplified the bias. So the MMAE was replaced with a Kalman filter to calculate sensor drift and a second Kalman filter to smooth the data. The sensor bias calculated by the filter was then subtracted from the measured signal before the measurement was sent to the second filter.

In the heart pump system, there are only position sensor measurements and all redundancy is eliminated by the time the measurements reach the controller. Instead, the plant model is used in place of a measured value in determining sensor bias. The recursive discrete Kalman filter is used, with:

$$\tilde{y}_k = \begin{bmatrix} Y_k \\ \Theta_k \end{bmatrix} = \begin{bmatrix} \int y_k dt \\ \int \theta_k dt \end{bmatrix}, \text{ where } y \text{ and } \theta \text{ are the model output, or simulated measurement}$$

values, and

$$x = \begin{bmatrix} y \\ \beta_y \\ \theta \\ \beta_\theta \end{bmatrix};$$

$$\Phi = \begin{bmatrix} 1 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \Gamma = \begin{bmatrix} \Delta t & 0 \\ 0 & 0 \\ 0 & \Delta t \\ 0 & 0 \end{bmatrix}; \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

The Q matrix was adjusted until the estimated bias converged quickly and accurately to the actual bias value.

An updated plant model and PD controller became available shortly before the sensor drift calculation filter system was implemented, and thus they were used for simulations of this new filtering scheme.

A bias calculating Kalman filter was added to the model and filter simulation file provided by Dr. A. Crassidis. This model only considers four of the eight states, y , $\dot{y}$, θ_x , and $\dot{\theta}_x$ (because of system symmetry, the other states should be the same), and the model output was used as feedback in the system; the noise and Kalman filter were added to the output. Two separate Kalman filters are used; one discrete-time steady-state and one continuous-time steady-state filter. Both are calculated using the built in MATLAB Kalman function.

The bias filter uses the position value output as the rate input and the integral of the position output from the ideal model for the measured value. Once it was confirmed that the filter could properly calculate the bias, the bias was subtracted from the Kalman Filter input. The gain calculating filter was then added, along with the error condition block, to allow faults to be introduced into the system. Like the noise, faults were also added to the system output and not fed back into the controller.

Once the bias filtering technique was demonstrated to be working well for all error conditions and for a sinusoidal bias (see appendix A.1 for the model output for each error condition), the system was modified to use the Kalman filter output as the feedback. The resulting system instability was determined to be a consequence of excess controller lag, so the controller had to be re-optimized. Once the new controller was available, the system was again tested for each error condition.

Covariance analysis was added to the system to determine when the integrated position error is outside of the three-sigma limits. The model was set up to switch the feedback for the system from the filtered ‘actual’ output to the ‘model’ output as the integrated position error goes out of bounds; once the bias calculations catch up with the ‘sensor error,’ the integrated position error will be within the bounds and the feedback will be switched back to the filtered ‘actual’ measurements. The Simulink model and model output for selected error conditions is shown in appendix A.2. In the final step, the remaining states, x , $\dot{x}$, θ_y , and $\dot{\theta}_y$ were added back into the system. The final model is shown below in Figure 12.

Figure 12: Final Simulink Model

4.0 Testing

Final testing of the fault-tolerance/fault detection system was carried out first through a simulation and then using the experimental test rig. Four types of errors were investigated for this work: unplugged sensor, sensor output equal to ground (short), sensor output equal to sensor input (short or sensor damage due to extreme over-power), and sensor bias. Sensor bias is the most frequently observed error for the HESAs used by the heart pump project. All error conditions and combinations of error conditions were tested for all sensors in simulation, however, only a few were selected for experimental testing because of limited test rig availability and inability to accurately test certain sensor malfunctions. Because of system symmetry, acceptable results for the selected faults should prove acceptable results for the remaining testable fault conditions.

4.1 Simulation

An ‘error block’ (Figure 13) was created for the simulation of faults. Both noise and any sensor fault effects were added at this point in the simulation. Since the plant model and simulation use only position values, the corresponding voltages first have to be calculated for each sensor from the ‘actual’ position, and then the voltage can be adjusted to match the current error state (i.e. set to zero for shorted to ground, etc.). Then the new (faulty) position values are calculated from the final voltage values.

Each error case was given a number, listed in Table 3, so to change the error condition, one only has to adjust the error condition number in the main simulation file. The error function can also accept arrays, so multiple errors can be introduced at once; a corresponding error time array is used to set the start time of each error.

The system model with fault options and Kalman filter was tested for various fault conditions, shown in Table 2, below, and compared to the behavior of the system with no filter.

Table 2: Considered Fault Types and Consequences

Fault	Consequence	Unfiltered Behavior	Filtered Behavior
Shorted to ground	Differenced position = actual distance $\pm \sim 1.75 \times 10^{-3}$ m	Unstable	Stable after initial settling time
Shorted to +5V signal	Differenced position = actual position $\pm \sim 7 \times 10^{-4}$ m	Unstable	Stable after initial settling time
Unplugged	Differenced position = actual position $\pm \sim 1.6 \times 10^{-3}$ m	Unstable	Stable after initial settling time
Sensor Bias	Differenced Position = actual position $\times (1 + \text{bias}) / 2$	Unstable	Stable after initial settling time
Sinusoidal Bias	Position = Position + Sinusoidal Error	Unstable	Stable after initial settling time
Bias increasing with time	Position = Position + Position $\times (\text{biasFactor} + 1) \times \text{time} / 2$	Unstable	Stable after initial settling time

Table 3: Fault Conditions

#	Fault
0	no faults
1	y_{fp} shorted to ground
2	y_{fp} shorted to +5V signal
3	y_{fp} unplugged
4	y_{fp} bias
5	y_{fn} shorted to ground
6	y_{fn} shorted to +5V signal
7	y_{fn} unplugged
8	y_{fn} bias
9	y_{rp} shorted to ground
10	y_{rp} shorted to +5V signal
11	y_{rp} unplugged
12	y_{rp} bias
13	y_{rm} shorted to ground
14	y_{rm} shorted to +5V signal
15	y_{rm} unplugged
16	y_{rm} bias
17	y_{rp} time-varying bias
18	x_{fp} shorted to ground
19	x_{fp} shorted to +5V signal
20	x_{fp} unplugged
21	x_{fp} bias
22	x_{fn} shorted to ground
23	x_{fn} shorted to +5V signal
24	x_{fn} unplugged
25	x_{fn} bias
26	x_{rp} shorted to ground
27	x_{rp} shorted to +5V signal
28	x_{rp} unplugged
28	x_{rp} bias
30	x_{rm} shorted to ground
31	x_{rm} shorted to +5V signal
32	x_{rm} unplugged
33	x_{rm} bias

Figure 13: Error Block

4.1.1 Results

Error Condition Zero: No Position Sensor Errors

Figure 14: Control Effort for y , θ_x , for System Operating with No Position Sensor Errors

Figure 15: Control Effort for x , θ_y , for System Operating with No Position Sensor Errors

As shown by Figures 14 and 15, the control effort required for a system with properly functioning components quickly settles to essentially zero, with noise. For the ideal model (without noise added), the controller effort is much smaller after the initial settling time, so the noise in the controller effort in Figures 14 and 15 demonstrates the additional effort expended by the controller due to the position noise.

Figure 16: Position and Velocity Output for y , θ_x , for System Operating with No Position Sensor Errors

Figure 17: Position and Velocity Output for x , θ_y , for System Operating with No Position Sensor Errors

Figures 16-19 show that the rotor is centered quickly, and other than some variation due to noise, remains centered. The radial and angular velocity plots demonstrate the utility of Kalman filtering in providing estimated velocity values; the measured value is the derivative of the (noisy) position value, and so is thus even noisier than the position data. The filtered estimate, however, has much less noise and so is very close to the model output. The rotation and displacement plots also show reduction in noise in the filtered output, but it is not as marked as in the case of the derivatives.

Figure 18: Bias Calculation Filter Output for y, θ_x , for System Operating with No Position Sensor Errors

Figure 19: Bias Calculation Filter Output for x, θ_y , for System Operating with No Position Sensor Errors

The error plots in Figures 18 and 19 show the error quickly converging to within the covariance bounds. The inner black lines represent the one- σ boundaries and the outer black lines represent the three- σ boundaries. As expected, most of the data falls within the three- σ boundaries, and calculations show that approximately 68% of the data falls within the one- σ boundaries. The bias is correctly calculated as zero, with some measurement noise.

Error Condition 1: y_{fp} Shorted to Ground

Figure 20: Control Effort for y , θ_x , for Error Condition 1, starting at $t=1s$

Figure 21: Control Effort for x , θ_y , for Error Condition 1, starting at $t=1s$

Figures 20 and 21 show that after an initial increase when the fault occurs, the control effort for this fault case is the same as for the case with no position sensor faults. In other words, the fault compensation (bias removal) works well enough that no extra effort is required on the part of the controller.

Figure 22: Position and Velocity Output for y , θ_x , for Error Condition 1, starting at $t=1$ s

Figure 23: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1s$

Figure 24: Position and Velocity Output for x , θ_y , for Error Condition 1, starting at $t=1s$

Figure 22 shows a large offset in the unfiltered y and θ_x output. Figure 23, the zoomed view, shows the raw output with the noise removed, the filtered output, and the model output; the bias is removed and after filtering the output is a close approximation to the model output. Figure 24 shows that the x and θ_y output is the same as for the no error case.

Figure 25: Bias Calculation Filter Output for y , θ_x , for Error Condition 1, starting at $t=1s$

The error plots in Figure 25 show the error spiking at $t=1s$, when the error first occurs, but then it is quickly brought back within bounds once the bias is estimated and subtracted from the output before filtering. Likewise, the measurement bias plots show the bias starts at zero and jumps at approximately $t=1s$ and then quickly settles to the constant bias value. Figure 26 shows the same x and θ_y error and bias output as for the case of no errors.

Figure 26: Bias Calculation Filter Output for x, θ_y , for Error Condition 1, starting at $t=1s$

Error Condition 2: y_{fp} Shorted to +5V Signal

Figure 27: Control Effort for y, θ_x , for Error Condition 2, starting at $t=1s$

Figure 28: Control Effort for x, θ_y , for Error Condition 2, starting at $t=1s$

Figure 29: Position and Velocity Output for y, θ_x , for Error Condition 2, starting at $t=1s$

Figure 30: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 2, starting at $t=1s$

Figure 31: Position and Velocity Output for x , θ_y , for Error Condition 2, starting at $t=1s$

Figure 29 shows the offset in the raw position, starting at $t=1s$ caused by the position sensor being shorted to the positive five volt signal (sensor output is then equal to five volts), compared to the filtered and unfiltered output with bias subtracted to the model output. Figure 30 is a closer view of the filtered and unfiltered output with the bias subtracted compared to the model output, showing that once the bias is subtracted and the output is filtered it provides a reasonable approximation of the model output. Figure 31 shows that, like for error condition one, the x and θ_y output is the same as for the case without any position sensor errors.

Figure 32: Bias Calculation Filter Output for y , θ_x , for Error Condition 2, starting at $t=1s$

Figure 32 shows that, like error condition 1, the bias introduced at $t=1s$ is quickly calculated and subtracted, bringing the error back within the covariance bounds. Figure 33 shows that the x and θ_y output is unaffected by the errors in the y direction position sensors.

Figure 33: Bias Calculation Filter Output for x, θ_y , for Error Condition 2, starting at t=1s

Error Condition 3: y_{fp} unplugged

Figure 34: Control Effort for y, θ_x , for Error Condition 3, starting at t=1s

Figure 35: Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1$ s

Figure 36: Zoomed View of Position and Velocity Output for y , θ_x , for Error Condition 3, starting at $t=1$ s

Figure 37: Bias Calculation Filter Output for y , θ_x , for Error Condition 3, starting at $t=1s$

Figure 38: Zoomed View of Bias Calculation Filter Output for y , θ_x , for Error Condition 3, starting at $t=1s$

Figures 34-38 show that, like the previous error conditions, the system quickly compensates for the sudden introduction of a large offset. The x and θ_y plots are also identical to all previous x and θ_y plots, and so are not shown here.

Error Condition 4: y_{fp} Bias with a Bias Factor of 0.99

Figure 39: Control Effort for y , θ_x , for Error Condition 4, starting at $t=1s$

Figure 40: Position and Velocity Output for y , θ_x , for Error Condition 4, starting at $t=1s$

Figure 41: Bias Calculation Filter Output for y, θ_x , for Error Condition 4, starting at t=1s

Figures 39-41 show the fault-tolerance method can also correct for a slight bias on one of the sensors, in the same manner it corrects for the other types of faults. Again in this case, the x and θ_y plots are also identical to all previous ones, and so are not shown here.

Error Condition 17: Bias on y_{rp} , increasing with time

The sensor output is multiplied by the bias factor. The bias factor is determined as follows:

$$BiasFactor = \begin{cases} 1 & t < t_{error} \\ 1 + (t - t_{error}) * 0.0001 & t \geq t_{error} \end{cases}$$

Figure 42: Control Effort for y, θ_x , for Error Condition 17, starting at $t=1s$

Figure 43: Position and Velocity Output for y, θ_x , for Error Condition 17, starting at $t=1s$

Figure 44: Bias Calculation Filter Output for y , θ_x , for Error Condition 17, starting at $t=1s$

Gradual sensor drift over time is one of the most likely fault conditions to occur in an actual system. Unlike the previous fault cases shown, this fault increases gradually over time instead of occurring suddenly, so there is no initial instability in the system.

Sinusoidal Noise Condition:

A sinusoidal signal with the following characteristics was added to all position sensor outputs:

Amplitude = 1×10^{-5} for the position and 1×10^{-3} for the angle

Frequency = 1 rad/sec

Figure 45: Control Effort for y, θ_x , with low frequency sinusoidal noise

Figure 46: Position and Velocity Output for y, θ_x , with low frequency sinusoidal noise

Figure 47: Bias Calculation Filter Output for y, θ_x , with low frequency sinusoidal noise

Figures 45-47 demonstrate the ability of the filter system to filter out low frequency periodic noise. Figures 48-50 show the limitations of the system to be able to react to rapidly changing bias.

High Frequency Sinusoidal Noise:

A sinusoidal signal with the following characteristics was added to all position sensor outputs:

Amplitude = 2.5×10^{-6} for position and 2.5×10^{-4} for angle

Frequency = 100 rad/sec

Figure 48: Control Effort for $y, \theta x$, with high frequency sinusoidal noise

Figure 49: Position and Velocity Output for $y, \theta x$, with high frequency sinusoidal noise

Figure 50: Bias Calculation Filter Output for y , θ_x , with low frequency sinusoidal noise

The bias graphs in Figure 50 show that while the bias is reasonably closely approximated, there is still a bit of lag between the actual and calculated bias. This lag appears as a lower amplitude periodic offset in the position graphs in Figure 49. These graphs show the limits of the ability of the bias estimation system to account for a rapidly changing fault condition.

4.1.2 Sensitivity to Plant Model Discrepancies

A sensitivity analysis was completed to determine the effects of discrepancies between the plant model and the ‘real’ model. Parameters used in the plant model calculation (see Appendix D) were adjusted individually for the actual pump plant model in the simulation while the original plant model was used for the Kalman filters. It was found that a difference of as little as 2% between any of the parameters used for these two plant models causes improper operation, as shown in Figure 51.

Figure 51: AMB spring constant adjusted by 5%, no error conditions.

The output shown in Figure 51 is not completely unstable, but differences of 10% or more cause errors in the Simulink model. For a more complex system such as the heart pump, it is not likely feasible to obtain sufficient plant model accuracy for the Kalman bias calculation system to be implemented successfully. Adjusting the model covariance value in the bias calculation filter improves the robustness of the system to modeling errors to some extent, but at the cost of bias convergence time.

4.2 Experimental Testing

Preliminary testing to determine whether the pump behaved similarly to the model when a fault was introduced (with no filtering or other fault-tolerance) was completed successfully on one of the heart pump lab's test pumps.

A few tests were run to confirm the findings of the sensitivity analysis. In addition to tests completed using the test rig, a 'pump compatible' simulation file was provided by the heart pump lab. This file is supposed to emulate the actual pump behavior well enough to provide an accurate prediction of whether the observer/filter will work on the actual pump.

To test the simulation on the actual system, the controller/filter system was implemented on a computer connected to the test rig. The effects of selected faults were also tested using the heart pump test rig, and then tested again with the addition of the Kalman filter. These

results were then compared to the results of the simulation in order to validate the simulation results.

The test procedure is outlined in appendix C.

4.2.1 Test Rig

The heart pump lab has a test rig which approximates the operation of a real heart pump for testing purposes. Most experiments are completed using air as the pumping fluid, but with additional components water or a mixture of water and glycerin can be used. The water-glycerin mixture provides the most accurate representation of human blood. While blood is non-Newtonian, at the speeds the pump operates at it behaves like a Newtonian fluid, so the Newtonian water-glycerin mixture is an appropriate substitute.

Figure 52: Levitation Test Rig Set-Up used for preliminary tests [4]

All experiments completed for this project used air as the pump fluid.

4.2.2 Preliminary Experimental Test Results

Preliminary experimental results were obtained by unplugging and shorting position sensors while the pump was operating. Each of these faults caused the rotor to be pulled to one wall as the controller tried to compensate for the faulty reading; this is the equivalent of the

position increasing exponentially in the Simulink simulation. The results for nominal operation and fault conditions 11, 20, and 27 are shown below in Figures 53-56.

Figure 53: Nominal Pump Behavior

During nominal operation, the rotor is essentially centered, with some noise apparent in the signal. While the noise in the x-direction sensor measurements appears to be generally random, there is a noticeable periodic component in the y-direction measurements. The signal noise was analyzed to check that the noise values selected for use in the Kalman filter were reasonable.

For the results shown in Figure 54, the rear positive y sensor was unplugged after two seconds of normal operation. The x-direction position/rotation shows very little response, while there is a clear offset in the y-direction position/rotation.

Figure 54: Rear positive y sensor unplugged (fault condition 11) experimental data

Figure 55: Rear positive x sensor shorted to +5V signal (error condition 27)

In Figure 55, the position sensor was shorted after approximately 1.6 seconds, resulting in a clear offset in the x-direction, but not in the y-direction. The intermittent change in bias (such as at $t=2s$) is due to an inconsistently maintained short.

Figure 56: Front positive x sensor unplugged (error condition 20)

Once the filter system was completed, some tests were completed on the heart pump test rig using the filter system as an observer only. Using the plant model output as input to the bias calculation filter, the observer calculated a significant sensor offset, even though the pump was operating properly with no position sensor faults. Output from this test is shown in Figure 57. An additional test was completed where the position sensor input was integrated and used in place of the integrated plant model output for the input to the bias calculation filter. Results from this test is shown in Figures 58 and 59. In this case, the while the observer calculates a bias, it is smaller and the observer correctly determines error to be within the covariance limits. However, because the same sensor input is used twice, the error would be within the covariance limits even in the case of a position sensor fault.

Figure 57: Observer Output

Figure 58: Observer Output

Figure 59: Rotor position output

4.2.3 LVADsim35 Results

Output from the ‘pump-compatible’ simulation is shown below.

Figure 60: y-position output from heart-pump emulator during normal pump operation. The output is essentially the same for the other position values.

Figure 61: Fault detection output from heart-pump emulator during normal pump operation.

As shown in Figure 60, the output immediately becomes unstable when the observer/filter is run with the pump emulator. Figure 61 shows that the fault detection system immediately identifies the (normal) pump operation as faulty. This incorrect identification explains the instability as shown in Figure 60; if there is a plant-model mismatch or other inconsistency between the pump emulator and the fault detection/compensation system, normal output will falsely trigger the compensation system, and as the pump emulator is not producing the results expected by the observer, attempts to correct for this fault will continually increase, resulting in system instability.

5.0 Interpretation of Results

The filter system outputs state estimates for the eight rotor states and four binary values indicating whether the Y and Tx error values are within the covariance limits, and whether the X and Ty error values are within the covariance limits. In addition, a binary value indicating whether the y/θ_x and x/θ_y values are within the acceptable limits for normal operation is returned. As the filter system compensates for the bias, the position error values reduce to within the covariance limits, so are not useful for determining whether a fault is present. The calculated bias value, however, provides a persistent indication of fault conditions, and could easily be used in the final system to activate an error alert system.

The simulation results show that the system works well for the case where the estimated plant model is identical to the actual pump behavior. After an error occurs, the bias calculating filter quickly determines the bias and the filtered data with the bias subtracted provides a good approximation of the rotor position. In the case of a bias which increases steadily with time or a sinusoidal error signal, there is a slight offset between the actual and calculated bias as it takes a few time-steps for the bias calculation filter to converge to the actual bias; when the bias is constantly changing, it cannot be calculated as accurately. Therefore, this type of filter is most useful for steady-state or slowly changing faults.

The Kalman filter effectively removes noise from the signal. During normal operation, the filter significantly reduces the noise in the estimated signals compared to the measured signals. The velocity and radial velocity are determined by taking the derivative of position and angle output, respectively, and thus are very noisy, so the affect is particularly apparent in these signals. The velocity and radial velocity output after Kalman filtering is significantly smoothed and much closer to the ideal model output.

The sensitivity analysis illustrates the inherent problems with using model-based fault detection; even small inaccuracies in the mathematical model of the system can make it impossible to properly detect and compensate for faults. In the simulation, differences between the estimated plant model and the ‘actual’ plant model of as little as two percent for one of the parameters causes improper operation and differences of ten percent or more result in errors in the model. The differences between the plant model and ‘actual’ plant model cause the fault detection system to consider proper operation to be faulty, because of the difference between the actual output and the estimated correct output. The estimator then attempts to account for that by adjusting the feedback to the controller; since the model is inaccurate, the adjustments will not correct the discrepancy, and so the feedback will be adjusted further. Since the adjustments are calculated using an inexact plant model, they are also incorrect, and the increasing adjustments will cause unstable operation of the system. Therefore, under these conditions, the estimator is essentially creating artificial position sensor fault conditions instead of compensating for them.

The results using the pump emulator and the test rig confirm these findings. In the case where the estimator is used as an observer only the behavior is a bit different since the incorrect position estimates are not fed back to the controller. In that case, because the adjustments are not affecting the output, the calculated bias and resulting position estimate increase exponentially, as shown by the tests using the test rig and test rig simulation file, but because it is not fed back into the system, it will continue to operate.

6.0 Conclusions

The lack of usable sensor redundancy limits the options for fault detection and fault-tolerance in the heart pump system. The Kalman filter approach was investigated because implementation does not require sensor redundancy. The results show that the Kalman filter with bias removal is not an ideal solution from a fault detection and fault-tolerance standpoint because it relies on an exact dynamic model of the system, which is typically not feasible.

The simulations illustrate that the solution works well provided that the dynamic model of the system is very accurate. If the model is inaccurate, the fault detection system will consider proper operation to be faulty, and in attempting to account for the discrepancy will essentially introduce an artificial position sensor error. The tests using the test rig and test rig simulation file confirm that, because of modeling inaccuracy, the estimator is likely unusable for a real system, especially a highly unstable one.

In addition to the difficulty in achieving sufficient modeling accuracy for this type of system, operating conditions add additional complications. The dynamic behavior of the pump changes based on rotational speed and patient orientation, and thus the plant model would have to be adjusted as well. Redundant sensors would allow for detection of faults, but not for proper fault compensation without a closer match between the estimated and actual plant model. Using the position sensor output before differencing would provide a level of redundancy which would allow the bias calculating filter to operate properly, however, the plant model could not be used to estimate the states when the error exceeds the acceptable limits as determined by the covariance analysis.

7.0 Future Recommendations

The addition of redundancy would significantly increase the methods available for fault detection and fault-tolerance, so a useful next step would be to explore self-sensing as a method to introduce redundancy into the system. With redundancy, a knowledge based solution would be feasible and relatively easy to implement, and could be used both to detect faults and failures and to compensate for them. Reconfigurable controllers, such as those which vary the number of position sensors used based on the fault condition, would also be worth exploring. The bias calculating Kalman filter based solution also might work much better in a system where redundant position information is available. Fault-tolerance systems using an H_∞ controller would also be worth investigating as an option for accounting for the varying operating conditions which would exist for an implanted heart pump.

Future projects could look at practical ways of implementing a pre-filtering system in addition to the Kalman filter in order to reduce the initial instability caused by a sudden sensor failure. Because most sensor faults drastically alter the sensor output, a pre-filtering system could detect measurements which are clearly faulty and remove that position sensor from calculation (i.e. use on positive or negative sensor instead of the difference between them). Adding this functionality to the differencing circuit or methods for multiplexing signals to reduce the number of wires required for transmission could be explored.

7.0 References

- [1] P. Garimella, B. Yao, "An adaptive robust framework for model-based state fault detection," in American Control Conference, 2006, June 2006, pp. 5692-97.
- [2] Xiaodong Zhang, T. Parisini, M.M. Polycarpou. "Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach." IEEE Transactions on Automatic Control, 49, 8 Aug., pp. 1259-74, 2004.
- [3] F.N. Chowdhury. "Fault Detection in Closed Loop Adaptive Systems," in 2006 IEEE Mountain Workshop on Adaptive and Learning Systems, Jul. 2006, pp. 160-64.
- [4] Aditi Khare. "Estimation and control of the pump pressure rise and flow from intrinsic parameters for a magnetically-levitated axial blood pump," MS Thesis, Rochester Institute of Technology, Rochester NY, 2008.
- [5] Uhn Joo Na, Alan Palazzolo. "The Fault-Tolerant Control of Magnetic Bearings With Reduced Controller Outputs." Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control, v 123, n 2, pp. 219-24, June 2001.
- [6] Matthew O. T. Cole, Patrick S. Keogh, Mehmet N. Sahinkaya, Clifford R. Burrows. "Towards fault-tolerant active control of rotor-magnetic bearing systems." Control Engineering Practice: UKACC Conference Control 2002, 12, 4, Apr., pp. 491-501, 2004.
- [7] Raoul Herzog, Philip Bühler, Conrad Gähler, René Larssonneur. "Unbalance Compensation Using Generalized Notch Filters in the Multivariable Feedback of Magnetic Bearings." IEEE Transactions on Control Systems Technology, 4, 5, Sept., p. 580-86, 1996.
- [8] P. Schroder, A. J. Chipperfield, P.J. Flemming, N. Grum. "Fault Tolerant Control of Active Magnetic Bearings." Proceedings of the 1998 IEEE International Symposium on Industrial Electronics, ISIE. Part 2, Jul, pp. 573-578, 1998.
- [9] Michael Demetriou. "Robust Adaptive Techniques for Sensor Fault Detection and Diagnosis," in Proceedings of the 37th IEEE Conference on Decision and Control, Dec. 1998, pp. 1143-48.
- [10] M. O. T. Cole, P. S. Keogh, C. R. Burrows. "Fault-tolerant control of rotor/magnetic bearing systems using reconfigurable control with built-in fault

detection.” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, v 214, n 12, pp 1445-65, 2000.

- [11] M. N. Sahinkaya, M. O. T. Cole, and C. R. Burrows. “Fault detection and tolerance in synchronous vibration control of rotor-magnetic bearing systems.” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 215, 12, pp. 1401-16, 2001.
- [12] D. Montie, E. Maslen. “Self-Sensing in Fault Tolerant Magnetic Bearings.” Transactions of the ASME, 123, Oct., pp.864-70, 2001.
- [13] Myounggyu D. Noh, Seong-Rak Cho, Jin-Ho Kyung, Seung-Kook Ro, Jong-Kweon Park. “Design and Implementation of a Fault-Tolerant Magnetic Bearing System for Turbo-Molecular Vacuum Pump.” IEEE/ASME Transactions on Mechatronics, v 10, n 6, Dec., pp. 626-31, 2005.
- [14] H. Xue, Jiang, J.G. “Fault Detection and Accommodation for Nonlinear Systems Using Fuzzy Neural Networks,” in IPEMC '06. CES/IEEE 5th International Power Electronics and Motion Control Conference, 2006., Aug. 2006 Volume 3, pp. 1-5.
- [15] Sung-Kuk Paek, Byeong-Cheol Park, Myounggyu D. Noh. “Design of Fault-Tolerant Inductive Position Sensor,” in 16th IEEE International Conference on Control Applications, 2007. 1-3 Oct. 2007, pp. 617-21.
- [16] M. Kinnaert, R. Hanus, and Ph. Arte. “Fault Detection and Isolation for Unstable Linear Systems,” in IEEE Transactions on Automatic Control, April 1995 v 40, n 4, pp. 740-2
- [17] S. Ravi and T. Jayasingh. “Self-validating control algorithm for plants with primary sensor failure using multiple estimators.” Proceedings of ISCP '02, 2002, pp. 1807-10.
- [18] Andras Varga. “Computational Issues in Fault Detection Filter Design,” Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA, December 2002, pp. 4760-5
- [19] A. Varga. “Numerically reliable methods for optimal design of fault detection filters.” Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005, pp. 2391-6
- [20] Alberto Cardoso and António Dourado. “Supervision and Fault-Tolerant Control: a Robust Model-based Approach,” Proceedings of the 2006 IEEE International

Conference on Control Applications Munich, Germany, October 4-6, 2006, pp. 1511-6.

- [21] J.X. Shen, D.M. Vilathgamuwa, K.J. Tseng, and W.K. Chan. "A Novel Compact PMSM with Magnetic Bearing for Artificial Heart Application," in Industry Applications Conference, 1999, Thirty-Fourth IAS Annual Meeting, Conference Record of the 1999 IEEE, Oct. 1999, vol. 2, pp. 1201-07.
- [22] J.X. Shen, K.J. Tseng, D.M. Vilathgamuwa, and W.K. Chan. "A Novel Compact PMSM with Magnetic Bearing for Artificial Heart Application," in *Industry Applications* 36, 4, Jul./Aug., pp. 1061-8, 2000.
- [23] Yohji Okada, Naoto Yamashiro, Kunihiro Ohmori, Toru Masuzawa, Takashi Yamane, Yoshiaki Konishi, and Satoshi Ueno. "Mixed Flow Artificial Heart Pump With Axial Self-Bearing Motor." *IEEE/ASME Transactions on Mechatronics*, 10, 6, Dec., pp. 658-65, 2005.
- [24] C. Barthod, G. Lemarquand. "Degrees of freedom control of a magnetically levitated rotor." *IEEE Transactions on Magnetics*, 31, 6, part 2, Nov., pp.4202-04 1995.
- [25] F. B. Zhou, R. Gerber, S. Whitley, T. Twardowski, D.J. Witts. "The Levitation of the SIXEP Pump Shaft using Active Magnetic Bearings." *IEEE Transactions on Magnetics*, 31, 6, Nov., pp. 4196-8, 1995.
- [26] Fangmin Li, Wenjun Xu. "The research and Design of Full Digital Magnetic Levitating Bearing Control and Amplifier System," in The Sixth World Congress on Intelligent Control and Automation, 2006, 2006, vol. 2, pp. 6279-82.
- [27] Scott A. Green, Kevin C. Craig. "Robust, Digital, Nonlinear Control of Magnetic-Levitation Systems." *Journal of Dynamic Systems, Measurement, and Control*, 120, Dec., pp. 488-95, 1998.
- [28] Ting-Jen Yeh and Ying-Jer Chung. "Sliding Control of Magnetic Bearing Systems," in Proceedings of the 2000 American Control Conference, June 2000, vol. 3, pp. 1622-26.
- [29] John L. Crassidis and John L. Junkins. Optimal Estimation of Dynamic Systems. Boca Raton: CRC Press. 2004
- [30] Robert Grover Brown and Patrick Y.C. Hwang. Introduction to Random Signals and Applied Kalman Filtering, 2nd ed. New York: John Wiley & Sons, Inc. 1992

- [31] Douglas C. Montgomery, George C. Runger, and Norma Faris Hubele. Engineering Statistics, 3rd ed. New York: John Wiley & Sons, Inc. 2004

Other Sources

1. Xian-Qun Liu, Hong-Yue Zhang, Jun Liu, and Jing Yang. "Fault Detection and Diagnosis of Permanent-Magnet DC Motor Based on Parameter Estimation and Neural Network." *IEEE Transactions on Industrial Electronics*, 47, 5, Oct., pp. 1021-30, 2000.
2. Olaf Moseler, Rolf Isermann. "Application of Model-Based Fault Detection to a Brushless DC Motor." *IEEE Transactions on Industrial Electronics*, 47, 5, Oct., pp. 1015-20, 2000.
3. Siddhartha Swarnakar, Siddhartha Mukopadhyay, Debaprasad Kastha. "Fault Detection and Remedial Strategies for Inter-Turn Short Circuit Faults in a Permanent Magnet Brushless DC Motor," in IEEE Indicon 2005 Conference, Dec., pp. 492-5, 2005.
4. J. A. Vázquez, E. H. Maslen, H.-J. Ahn, D.-C. Han. "Model Identification of a Rotor with Magnetic Bearings." *Journal of Engineering for Gas Turbines and Power*, 125, Jan., p. 149-55, 2003.
5. Brad Paden, Nelson Groom, James F. Antaki. "Design Formulas for Permanent Magnet Bearings." *Journal of Mechanical Design*, 125, Dec, pp. 734-8. 2003.
6. Alexandre Schammass, Raoul Herzog, Philipp Bühler, Hannes Bleuler. "New Results for Self-Sensing Active Magnetic Bearings Using Modulation Approach." *IEEE Transactions on Control Systems Technology*, 13, 4, July, pp. 509-16, 2005.
7. Eric H. Maslen, Dominick T. Montie, Tetsuya Iwasaki. "Robustness Limitations in Self-Sensing Magnetic Bearings." *Journal of Dynamic Systems, Measurement, and Control*, 128, 2, June, pp. 197-203, 2006.
8. Roberto Bassani. "A Stability Space of a Magnetomechanical Bearing." *Journal of Dynamic Systems, Measurement, and Control*, 129, Mar., pp. 178-81, 2007.
9. Michael Baloh, Gang Tao, Paul Allaire. "Adaptive Estimation of Magnetic Bearing Parameters," in Proceedings of the 1999 IEEE International Conference on Control Applications, 22-27 August 1999, pp. 1193-98.

10. Zarchan, Paul. **Fundamentals of Kalman Filtering: A Practical Approach**, 2nd ed. Reston, VA. American Institute of Aeronautics and Astronautics. 2005.
11. Wyffels, Kevin L. “Development of a ground truth simulator and application of a generalized multiple-model adaptive estimation approach to tune a state estimation filter,” MS Thesis, Rochester Institute of Technology, Rochester NY, 2007.
12. Zhizhou Zhang, Zhiqiang Long, Longhua She and Wensen Chang. “Linear Quadratic State Feedback Optimal Control against Actuator Failures.” Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, August 5 - 8, 2007, pp. 3349-54

Appendices

Appendix A: Simulink Models and Associated Filter Outputs

A.1: Initial Model with Gain Calculation Kalman Filter, and Results

Figure 62: Postion and Velocity Output: No Errors

Figure 63: Bias and Integral of Position: No Errors

Figure 64: Position and Velocity Output: Error Condition 1, starting at t = 1s

Figure 65: Position and Velocity Output, Zoomed View: Error Condition 1, starting at $t = 1$ s

Figure 66: Bias and Integral of Position: Error Condition 1, Starting at $t = 1$ s

Figure 67: Bias and Integral of Position, Zoomed View: Error Condition 1, Starting at t = 1s

Figure 68: Position and Velocity Output: Error Condition 2, starting at t = 1s

Figure 69: Position and Velocity Output, Zoomed View: Error Condition 2, starting at $t = 1$ s

Figure 70: Bias and Integral of Position: Error Condition 2, Starting at $t = 1$ s

Figure 71: Bias and Integral of Position, Zoomed View: Error Condition 2, Starting at $t = 1$ s

Figure 72: Position and Velocity Output: Error Condition 3, starting at $t = 1$ s

Figure 73: Position and Velocity Output, Zoomed View: Error Condition 3, starting at $t = 1$ s

Figure 74: Bias and Integral of Position: Error Condition 3, Starting at $t = 1$ s

Figure 75: Bias and Integral of Position, Zoomed View: Error Condition 3, Starting at $t = 1$ s

Figure 76: Position and Velocity Output: Error Condition 4, starting at $t = 1$ s

Figure 77: Position and Velocity Output, Zoomed View: Error Condition 4, starting at $t = 1$ s

Figure 78: Bias and Integral of Position: Error Condition 4, Starting at $t = 1$ s

Figure 79: Bias and Integral of Position, Zoomed View: Error Condition 4, Starting at $t = 1$ s

Figure 80: Position and Velocity Output: Error Condition 5, starting at $t = 1$ s

Figure 81: Position and Velocity Output, Zoomed View: Error Condition 5, starting at $t = 1$ s

Figure 82: Bias and Integral of Position: Error Condition 5, Starting at $t = 1$ s

Figure 83: Bias and Integral of Position, Zoomed View: Error Condition 5, Starting at $t = 1$ s

Figure 84: Position and Velocity Output: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error

Figure 85: Position and Velocity Output, Zoomed View: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error

Figure 86: Bias and Integral of Position: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error

Figure 87: Bias and Integral of Position, Zoomed View: Error Condition 4, Starting at $t = 1$ s, Plus Sinusoidal Error

A.2: Four-State Model with Gain Calculation Kalman Filter, Covariance Analysis, and Feedback Switch, and Results

Figure 88: Final Simulink Model For $y, \dot{y}, \theta_x$, and $\dot{\theta}_x$ only

Figure 89: Bias Calculation Filter Output for System Operating with No Position Sensor Errors

Figure 90: Position and Velocity Output for System Operating with No Position Sensor Errors

Figure 91: Bias Calculation Filter Output for System Operating with No Position Sensor Errors (Noise Only)

Figure 92: Control Effort for Error Condition 1, starting at t=1s

Figure 93: Position and Velocity Output for Error Condition 1, starting at t=1s

Figure 94: Zoomed view of Position and Velocity Output for Error Condition 1, starting at t=1s

Figure 95: Bias Calculation Filter Output for Error Condition 1, starting at t=1s

Figure 96: Zoomed View of Bias Calculation Filter Output for Error Condition 1, starting at t=1s

Figure 97: Control Effort for Error Condition 2, starting at t=1s

Figure 98: Position and Velocity Output for Error Condition 2, starting at $t=1s$

Figure 99: Zoomed View of Position and Velocity Output for Error Condition 2, starting at $t=1s$

Figure 100: Bias Calculation Filter Output for Error Condition 2, starting at $t=1s$

Figure 101: Zoomed View of Bias Calculation Filter Output for Error Condition 2, starting at $t=1s$

Figure 102: Control Effort for Error Condition 3, starting at $t=1s$

Figure 103: Zoomed View of Control Effort for Error Condition 3, starting at $t=1s$

Figure 104: Position and Velocity Output for Error Condition 3, starting at $t=1s$

Figure 105: Zoomed View of Position and Velocity Output for Error Condition 3, starting at $t=1s$

Figure 106: Bias Calculation Filter Output for Error Condition 3, starting at t=1s

Figure 107: Zoomed View of Bias Calculation Filter Output for Error Condition 3, starting at t=1s

Figure 108: Control Effort for Error Condition 4, starting at t=1s

Figure 109: Zoomed View of Control Effort for Error Condition 4, starting at $t=1s$

Figure 110: Position and Velocity Output for Error Condition 4, starting at $t=1s$

Figure 111: Zoomed View of Position and Velocity Output for Error Condition 4, starting at $t=1s$

Figure 112: Bias Calculation Filter Output for Error Condition 4, starting at $t=1s$

Figure 113: Error Condition 17, starting at $t=1s$

Figure 114: Position and Velocity Output for Error Condition 17, starting at $t=1s$

Figure 115: Bias Calculation Filter Output for Error Condition 17, starting at $t=1\text{s}$

Appendix B: Individual Sensor Equations

Position equations for the differenced sensor output were provided by the RIT heart pump project. Equations for determining position using the output from a single sensor were determined as follows. As the equations vary by sensor, the values for the sensors on the heart pump at the time the data used was taken were used. Therefore, the sensors were assumed to be typical, or similar to any other sensor of the same time; while the simulation therefore does not provide an exact representation of any sensor that could be used in the heart pump, it provides a reasonable approximation.

Position data for nominal operation was provided courtesy of the Heart Pump Lab. Individual sensor outputs were plotted vs. the differenced voltages, and a best-fit curve was determined for each. Then, because the position equations for the differenced voltages assume that the slopes and intercepts of each are the same, the average was taken for each pair of sensors.

Appendix C: Position Sensor Fault Emulation Test Procedure

For each test, measure:

- Rotor Position (x, y, θ_x, θ_y)
- Current
- Force

At 5000 Hz.

Front positive y-direction position sensor, y_{fp} :

Test 1: Output = Zero

1. Operate pump until steady-state operation is achieved.
2. Begin recording data
3. Use a jumper wire to short the output of the rear positive x-direction position sensor to ground.
4. Continue running pump for two seconds or until steady-state operation is achieved.
5. Stop recording data.

Test 2: Output = Input

1. Operate pump until steady-state operation is achieved.
2. Begin recording data
3. Use a jumper wire to short the output of the rear positive x-direction position sensor to the sensor input.
4. Continue running pump for two seconds or until steady-state operation is achieved.
5. Stop recording data.

Test 3: Output = floating voltage

1. Operate pump until steady-state operation is achieved.
2. Begin recording data
3. Unplug the power to the rear positive x-direction position sensor.
4. Continue running pump for two seconds or until steady-state operation is achieved.
5. Stop recording data.

Front negative y-direction position sensor, y_{fn} :

Test 4: Output = Zero

Repeat Test 1 using front negative y-direction position sensor

Rear positive y-direction position sensor, y_{rp} :

Test 5: Output = Zero

Repeat Test 1 using rear positive y-direction position sensor

Rear negative y-direction position sensor, y_{rn} :

Test 6: Output = Zero

Repeat Test 1 using rear negative y-direction position sensor

Front positive x-direction position sensor, x_{fp} :

Test 7: Output = Zero

Repeat Test 1 using front positive x-direction position sensor

Rear negative x-direction position sensor, x_m :

Test 8: Output = Zero

Repeat Test 1 using rear negative x-direction position sensor

Test 9: Output = Input

Repeat Test 2 using rear negative x-direction position sensor

Test 10: Output = floating voltage

Repeat Test 3 using rear negative x-direction position sensor

Rear positive and negative x-direction position sensors, x_{rp} and x_{np} :

Test 11: Output = Zero

1. Operate pump until steady-state operation is achieved.
2. Begin recording data.
3. Use a jumper wire to short the output of BOTH the rear negative and positive x-direction position sensor to ground.
4. Continue running pump for two seconds or until steady-state operation is achieved.
5. Stop recording data.

Test 12: Output = Input

1. Operate pump until steady-state operation is achieved.
2. Begin recording data.
3. Use a jumper wire to short the output of BOTH the rear negative and positive x-direction position sensor to the sensor input.
4. Continue running pump for two seconds or until steady-state operation is achieved.
5. Stop recording data.

Test 13: Output = Input

1. Run pump until steady-state operation is achieved.
2. Begin recording data
3. Use a jumper wire to short the output of the rear negative x-direction position sensor to the sensor input.
4. Allow to run for 2 seconds.
5. Use a jumper wire to short the output of the positive x-direction position sensor to the sensor input.
6. Continue running pump for two seconds or until steady-state operation is achieved.
7. Stop recording data.

Appendix D: Plant Model Derivation

Dynamic Models:

x-direction

y-direction

Assumptions: All angles are small: $\sin \theta \approx \theta$ and $\cos \theta \approx 1$
Independent direction (i.e. x and y direction behavior can be separated)

Solution:

For the y direction:

where:

$$F_{k_1} = k_1(y - d_1 \sin \theta_x)$$

$$F_{k_2} = k_2(y + d_2 \sin \theta_x)$$

$$F_{k_3} = k_3(y - d_3 \sin \theta_x)$$

With the small angle assumption these reduce to:

$$F_{k_1} = k_1(y - d_1 \theta_x)$$

$$F_{k_2} = k_2(y + d_2 \theta_x)$$

$$F_{k_3} = k_3(y - d_3 \theta_x)$$

Summation of forces in the y-direction:

$$\begin{aligned} \sum F_y = m\ddot{y} &= F_1 + F_2 - F_{k_1} - F_{k_2} - F_{k_3} \\ &= F_1 + F_2 - k_1(y - d_1 \theta_x) - k_2(y + d_2 \theta_x) - k_3(y - d_3 \theta_x) \end{aligned} \quad (D1)$$

Combining terms:

$$m\ddot{y} = F_1 + F_2 - (k_1 + k_2 + k_3)y + (k_1 d_1 - k_2 d_2 + k_3 d_3)\theta_x \quad (D2)$$

Solving (D2) for $\ddot{y}$ yields:

$$\boxed{\ddot{y} = \frac{1}{m}F_1 + \frac{1}{m}F_2 - \frac{(k_1 + k_2 + k_3)}{m}y + \frac{(k_1 d_1 - k_2 d_2 + k_3 d_3)}{m}\theta_x} \quad (D3)$$

Summation of moments around the x-axis:

$$\begin{aligned}\sum M_{\theta_x} &= J\ddot{\theta} = -d_1 F_1 + d_2 F_2 + d_1 F_{k_1} - d_2 F_{k_2} + d_3 F_{k_3} \\ &= -d_1 F_1 + d_2 F_2 + d_1 k_1 (y - d_1 \theta_x) - d_2 k_2 (y + d_2 \theta_x) + d_3 k_3 (y - d_3 \theta_x)\end{aligned}\quad (D4)$$

Combining terms:

$$J\ddot{\theta} = -d_1 F_1 + d_2 F_2 + (k_1 d_1 - k_2 d_2 + k_3 d_3) y - (k_1 d_1^2 + k_2 d_2^2 + k_3 d_3^2) \theta_x \quad (D5)$$

Solving (D5) for $\ddot{\theta}$ yields:

$$\ddot{\theta} = -\frac{d_1}{J} F_1 + \frac{d_2}{J} F_2 + \frac{(k_1 d_1 - k_2 d_2 + k_3 d_3)}{J} y - \frac{(k_1 d_1^2 + k_2 d_2^2 + k_3 d_3^2)}{J} \theta_x \quad (D6)$$

For the x direction:

where:

$$\begin{aligned}F_{k_1} &= k_1 (x - d_1 \sin \theta_y) \\ F_{k_2} &= k_2 (x + d_2 \sin \theta_y) \\ F_{k_3} &= k_3 (x + d_3 \sin \theta_y)\end{aligned}$$

Using the small angle assumption, these reduce to:

$$\begin{aligned}F_{k_1} &= k_1 (x - d_1 \theta_y) \\ F_{k_2} &= k_2 (x + d_2 \theta_y) \\ F_{k_3} &= k_3 (x + d_3 \theta_y)\end{aligned}$$

Summation of forces in the x-direction:

$$\begin{aligned}\sum F_x &= m\ddot{x} = F_1 + F_2 - F_{k_1} - F_{k_2} - F_{k_3} \\ &= F_1 + F_2 - k_1 (x - d_1 \theta_y) - k_2 (x + d_2 \theta_y) - k_3 (x + d_3 \theta_y)\end{aligned}\quad (D7)$$

Combining terms:

$$m\ddot{x} = F_1 + F_2 - (k_1 + k_2 + k_3)x + (k_1d_1 - k_2d_2 - k_3d_3)\theta_y \quad (D8)$$

Solving (D8) for $\ddot{x}$ yields:

$$\boxed{\ddot{x} = \frac{1}{m}F_1 + \frac{1}{m}F_2 - \frac{(k_1 + k_2 + k_3)}{m}x + \frac{(k_1d_1 - k_2d_2 - k_3d_3)}{m}\theta_y} \quad (D9)$$

Summation of moments around the y-axis:

$$\begin{aligned} \sum M_{\theta_x} &= J\ddot{\theta} = -d_1F_1 + d_2F_2 + d_1F_{k_1} - d_2F_{k_2} - d_3F_{k_3} \\ &= -d_1F_1 + d_2F_2 + d_1k_1(x - d_1\theta_y) - d_2k_2(x + d_2\theta_y) - d_3k_3(x + d_3\theta_y) \end{aligned} \quad (D10)$$

Combining terms:

$$J\ddot{\theta} = -d_1F_1 + d_2F_2 + (k_1d_1 - k_2d_2 - k_3d_3)x - (k_1d_1^2 + k_2d_2^2 + k_3d_3^2)\theta_y \quad (D11)$$

Solving (D11) for $\ddot{\theta}$ yields:

$$\boxed{\ddot{\theta}_y = -\frac{d_1}{J}F_1 + \frac{d_2}{J}F_2 + \frac{(k_1d_1 - k_2d_2 - k_3d_3)}{J}x - \frac{(k_1d_1^2 + k_2d_2^2 + k_3d_3^2)}{J}\theta_y} \quad (D12)$$

Now, equations (D3), (D6), (D9) and (D12) are put into state-space form, where state-space form is:

$$\dot{z} = Az + Bu \quad (D13)$$

where z is the system states and u is the system input.

Since:

$$z = \begin{bmatrix} x \\ \dot{x} \\ \theta_y \\ \dot{\theta}_y \\ y \\ \dot{y} \\ \theta_x \\ \dot{\theta}_x \end{bmatrix} \quad (D14)$$

and $\dot{z}$ is the time derivative of z ,

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta}_y \\ \ddot{\theta}_y \\ \dot{y} \\ \ddot{y} \\ \dot{\theta}_x \\ \ddot{\theta}_x \end{bmatrix} \quad (D15)$$

Since this is an undamped system, there are no terms for $\dot{x}$, $\dot{\theta}_y$, $\dot{y}$, or $\dot{\theta}_x$ in any of the equations so far. These terms are calculated both in the Simulink model and actual heart pump by taking the time-derivatives of the radial and rotational position, and so the plant model output should be the same as the input for these values. Thus the four remaining equations are:

$$\dot{x} = \dot{x} \quad (D16)$$

$$\dot{\theta}_y = \dot{\theta}_y \quad (D17)$$

$$\dot{y} = \dot{y} \quad (D18)$$

$$\dot{\theta}_x = \dot{\theta}_x \quad (D19)$$

Accordingly,

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{(k_1 + k_2 + k_3)}{m} & 0 & \frac{(k_1 d_1 - k_2 d_2 - k_3 d_3)}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{(k_1 d_1 - k_2 d_2 - k_3 d_3)}{J} & 0 & -\frac{(k_1 d_1^2 + k_2 d_2^2 + k_3 d_3^2)}{J} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{(k_1 + k_2 + k_3)}{m} & 0 & \frac{(k_1 d_1 - k_2 d_2 + k_3 d_3)}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{(k_1 d_1 - k_2 d_2 + k_3 d_3)}{J} & 0 & -\frac{(k_1 d_1^2 + k_2 d_2^2 + k_3 d_3^2)}{J} & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{m} & \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{d_1}{J} & \frac{d_2}{J} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & \frac{1}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{d_1}{J} & \frac{d_2}{J} \end{bmatrix} \begin{bmatrix} F_{1x} \\ F_{2x} \\ F_{1y} \\ F_{2y} \end{bmatrix} \quad (D14)$$