FAULT DETECTION AND IDENTIFICATION OF A SPECIAL CLASS OF GENERALIZED STATE-SPACE SYSTEMS

by

Yaang (Ken) Zhao

B.Eng., Xi'an Jiaotong University, 1984

MBA, The University of Alberta, 1988

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the School

of

Engineering Science

SIMON FRASER UNIVERSITY

May 1997

0-612-24281-1

APPROVAL

Name: Yaang (Ken) Zhao

Degree: Master of Applied Science (Electrical/Control Engineering)

Title of Thesis: Fault Detection and Identification of a Special Class of Generalized State-Space Systems

Examining Committee:

Chair: Dr. Andrew Rawicz

Prof. Mehrdad Saif

Senior Supervisor

Prof. William Gruver

Professor Emeritus George Bojadziev

Date Approved: ______________________

Abstract

The focus of this thesis is on fault detection and identification of constrained mechanical systems. This kind of system can not be exclusively described by dynamic equations because the constraints represent algebraic relations among certain system variables. Such systems which are partially dynamic and partially algebraic are called generalized state-space systems, descriptor systems or singular systems. Constrained mechanical systems are a special class of descriptor systems because they lack infinite observability and complete controllability, which are desirable system properties for state estimator based fault detection methods. This thesis deals with the unique characteristics of constrained mechanical systems and presents a systematic approach for fault detection and control of such systems under uncertainties. In this thesis, actuator faults are modeled as unknown inputs to the dynamic equations of typically nonlinear constrained mechanical system. Sensor faults are added to the output equations of the system. The nonlinear system model is first linearized about an operating point. Then a coordinate transformation technique is used to convert the resultant linear descriptor form representation of the system into two sub-systems: a dynamic subsystem plus an algebraic subsystem. Based on the dynamic subsystem representation, an unknown input observer is designed to provide estimates of displacements, velocities, constraint forces, and sensor faults simultaneously. The estimates of sensor faults provide immediate means for

sensor fault detection and identification. The estimates of displacements, velocities, and constraint forces can be used in state feedback control of the system. Actuator fault detection and identification is accomplished by estimating actuator faults using a least square solution technique which uses the estimation of the state vector of the system. This model-based analytical redundancy approach offers many advantages. It can detect a wide variety of faults. It generates not only the magnitude but also the shape of the faults and thus possesses the capability of distinguishing between momentary faults and persistent ones. Moreover, Its mathematical simplicity and computational efficiency makes it a better candidate for computer simulation and/or real-time implementation. Simulation performed using a practical system (an UMS-2 robot) model indicates that the proposed approach is capable of detecting and identifying multiple and/or simultaneous actuator faults and sensor faults almost immediately.

Dedication

To the fountain of civilization - applied science and the people that love me or believe in me in my life

Acknowledgments

First, I would very much like to thank Dr. Mehrdad Saif, my supervising professor, for guiding me through my program of study, finding potential thesis topics for me, and helping me overcome some of the difficulties encountered in this thesis. His suggestions, comments, and corrections with respect to this thesis are very much appreciated.

Secondly, I would like to thank Nancy, my loving wife, for her constant and strong support.

Thirdly, I would like to thank Professor William Gruver and Professor Emeritus George Bojadziev for serving on my thesis committee and spending their valuable time reviewing the thesis. I would also like to thank Dr. Andrew Rawicz, Chairman of the thesis examining committee, for his time.

Fourthly, I would like to thank Ms. Brigitte Rabold, Graduate Secretary of the School of Engineering Science at SFU, for the excellent services she provided during my program of study.

Fifthly, I would like to thank my friend Yidong Zhang for advising me to use Microsoft Word to write the thesis and acting as a communication agent between me and SFU while I was doing the last part of my thesis off campus.

Contents

List of Figures

Chapter 1

Introduction to Fault Detection and Identification

Automatic systems have been widely employed in commerce and industry for many years. Technological progress has made many of these systems more complex and sophisticated. Examples of these dynamic systems include commercial and military aircraft, navigation systems, space shuttle, nuclear reactors, chemical reactors, robots, and many others. These systems can consist of many working parts which may malfunction or fail at any time. Complete failure of these systems, especially those mission-critical ones, can result in unacceptable economical loss and/or human casualty. The need for reliability and fault-tolerance in these systems at reasonable cost prompted and in some cases fueled research in fault detection, isolation, and accommodation. New developments in fault diagnosis of dynamic systems started to appear in the 1970's. Some basic theoretical and application results were achieved in the 1980's and early 1990's. Research in this complex, diverse, and relatively new field continues today and is expected to continue well into the distant future. For example, several major aircraft manufacturers and car makers currently have some kinds of their own R&D activities in this area. On-board fault detection or registration may become a design criterion in some models of airplanes and automobiles.

A dynamic system, or a plant as it is commonly referred to, can be divided into three types of subsystems: actuators, main structure or process (which may consist of components), and instrumentation/sensors. Let's take an aircraft flight control system as an example. The actuators are the servomechanism that drive the control surfaces and engines which provide the driving thrust. The autopilot controller provides the actuators with the input or control signals. The main structure is the airframe with its cargo and appendages, along with the aerodynamic forces exerted on the control surfaces. The instrumentation consists of several sensors or transducers attached to the airframe. The sensors provide signals proportional to the vital motions of the airframe. These signals include airspeed, altitude, heading, acceleration, attitude, rate of change of attitude, control surface deflection, engine thrust, etc.. Sensor signals are fed back to the autopilot which uses the feedback information and reference/command inputs in its dynamic determination of new control signals. The actuators execute the new control signals dynamically and possibly affect the state of the system and sensor measurements again. Such a system is called a closed-loop control system or a feedback control system in control system engineering.

Research in the field of fault diagnosis has led to the invention of some jargons. Three of the most commonly used ones are FDI (Fault Detection and Isolation/Identification), FDIA (Fault detection, Isolation/Identification, and Accommodation), and IFD (Instrument Fault Detection). Fault detection and identification means declaring the occurrence of faults and indicating which sensors, actuators, or components are faulty. Fault accommodation refers to the reconfiguration of system signals or component actions in order to permit continued operation of the system. Fault accommodation is an application-specific task and is not addressed by most researchers. Another thing that has not been addressed by most researchers in FDIA is the reliability of typically digital computers, which are usually used in the implementation of FDI algorithms. Research in fault diagnosis have been focused on sensor fault detection and to a lesser extent on actuator fault detection, although one research work using least square parameter estimation methods has shown its capability of detecting and localizing process faults or component failures. A typical fault monitoring scheme is usually designed to detect and correct faults in one or two of the three subsystems. Early proposed schemes were primarily concerned with sensor fault detection. Once detected, sensor faults could usually be corrected by electronic switching techniques and do not require the reconfiguration of mechanical parts. On the other hand, actuator fault accommodation is usually more difficult than re-directing electrical signals. The compensation of faults in the main structure is even less feasible and usually requires expert knowledge of the underlying system. This is probably one of the most challenging aspects of any practical FDIA scheme.

The traditional approach to fault tolerance in dynamic systems is hardware redundancy. Typically three or four identical or similar hardware elements (actuators, measurement sensors, process components, etc.) are distributed spatially around the system to provide protection against localized damage. Multiple elements are used to perform a single task for which one element is sufficient if it was completely reliable. For example, three or more sensors could be installed to measure the same output. The measurements from the sensors could be compared in a logic circuit for consistency. If the measurement from one sensor deviates too much from the average of the measurements from the other sensors, then this sensor is declared faulty. The underlying reasonable assumption is that the other sensors remain within a small difference from each other. Additionally, the logic circuit gives some allowances for electronic noise, manufacturing tolerance, and monitoring errors inherent in instruments. The hardware redundancy approach is generally simple and straightforward to apply. It is therefore widely used. It is essential in the control of aircraft, space vehicles and in certain safety-critical process plants that involve nuclear reactors or dangerous chemicals.

The major problems associated with hardware redundancy or physical redundancy are the extra cost and software and, furthermore, the additional space required to accommodate the redundant equipment and/or the extra weight brought on by the redundant equipment. In aircraft, for example, the additional space could be used for more mission-oriented equipment. The additional weight limits the pay-load for defensive equipment and, most particularly, for fuel. Moreover, since redundant sensors tend to have similar life expectancies, it is likely that when one sensor fails the other will soon become faulty too.

New developments in FDIA have been prompted by the high cost of excess hardware and the space and weight penalties associated with hardware redundancy since the early 1970's. The availability of reliable and powerful computers also contributed to the developments of new approaches which eliminate some or all of the redundant hardware. These new approaches to FDIA are based on functional redundancy inherent in the systems. The fundamental idea is that entirely different measurements from three (or more) dissimilar sensors are driven by the same dynamic state of the system and are therefore functionally related. These different signals can be used in a comparison scheme more sophisticated than the simple majority-vote logic used in hardware redundancy approaches to detect and identify sensor faults. These newer schemes were initially called inherent redundancy or functional redundancy to distinguish them from physical or hardware redundancy. They are now better or alternatively known as analytical redundancy or artificial redundancy. Virtually all of the published research works in FDIA belong to this new class of approaches, although it has been recognized that both hardware redundancy and analytical redundancy approaches can be and in many cases should be employed together to advantage.

Analytical redundancy can use and has used knowledge from several academic disciplines. These include but are not limited to control theory, statistics, and computer science. Specific techniques employed in analytical redundancy FDI approaches include state estimation, parameter estimation, adaptive filtering, variable threshold logic, statistical decision theory, and combinatorial and logic operations. There are plenty books and papers on these disciplines and subjects. For example, the book of Swisher (1976) and the book of Chen (1984) contain information on reduced-order and full-order observer design techniques which can be used for state estimation. Other basic concepts in control theory such as state-space modeling, state (variables), state controllability, output controllability, observability, state feedback, output feedback, and stability are also covered in these books. The book of Dai (1989) provides singular control system theory which is useful in dealing with generalized state-space or descriptor systems among which are constrained dynamic systems.

All of the aforementioned techniques can be implemented using high speed digital computers or electrical circuits. High level system simulation or modeling languages such as MATLAB, Simulink, or MatrixX can be used in simulation of FDI schemes on dynamic systems. Lower level languages such as Assembly or C can be used in experimentation or real-time application.

Analytical redundancy FDI approaches are essentially based on modeling dynamic systems in one way or another. Either the dynamic nature of the system is known to a reasonable degree of precision or the physical parameters of the system can be determined by some kinds of on-line identification techniques. Normally the FDI subsystem is constructed in parallel to the monitored system. It can use both the input signals and the output signals of the monitored system to generate signals within itself. These generated signals serve the same purpose as the majority-vote signals used in hardware redundancy, i.e. they can be used in logic processing or other kinds of sophisticated algorithm to detect faults and identify faulty elements.

To illustrate the basic notion of IFD scheme, assume that there are p sensors and one of the them is known to be reliable. Also assume that an observer or state estimator can be constructed using the measurement signal from this reliable sensor and the inputs to the monitored plant. In this case the FDI subsystem can generate estimates of the measurement signals of all the other sensors. These estimates can then be compared with their actual counterparts. Simple threshold logic can be applied to the difference signals to detect and identify sensor faults. In view of the noise in sensor signals and the inaccuracy in system modeling and estimation, the thresholds shall be non-zero to prevent false alarms and yet small enough to allow the FDI scheme remain sensitive to moderate faults. Obviously, there is a compromise or balance between sensitivity to incipient (slowly developing or small) faults and false alarm rate in this case as in many other cases. Incidentally, this example is known as dedicated observer scheme (DOS), which was presented by Clark (1979). Many variations and alterations of this simple idea are possible.

Functionally-redundant FDI schemes may be further classified into at least three sub-classes according to the techniques used in the schemes. The first sub-class of schemes uses state estimation technique which is believed to be the most widely employed technique in all analytical redundancy FDI schemes. This technique is suitable for systems for which a set of differential equations (plus a few algebraic equations in the case of constrained dynamic systems) can be fairly easily obtained by applying the physical or engineering laws governing the motions of the system. Examples of such systems can include aircraft and robots. The approach presented in this thesis falls into this sub-class. Typically the nonlinear mathematical model of the dynamic system in this sub-class is linearized and also converted into a state-space representation format. The analysis of the system and the design of state estimator or observer based FDI subsystem is carried out in the realm of linear system theory, or linear singular system theory in the case of constrained dynamic systems. The second sub-class of analytical redundancy approaches uses parameter estimation techniques. A survey of the schemes in this sub-class is presented in the paper of Isermann (1984). A thoroughly studied method in this sub-class is the so called least-squares parameter estimation technique. This approach can provide on-line estimates of physical system parameters. Estimated parameters associated with specific subsystems of the plant or process can be used to detect and identify faults in these subsystems or components. This method is considered to be particularly important for process plants such as chemical processes and nuclear reactors. In these process plants, parameter variations result from process faults can cause rapid parameter estimate changes, even though the process itself typically has a slow dynamic behavior. This approach can detect and identify both component faults and sensor faults. The third sub-class of analytical redundancy based approaches uses the so called parametric modeling technique. A parametric model is essentially an estimator of a process variable using other process variables as inputs. Some simulation and actual experiments using this approach have been performed at several nuclear power stations. Readers who are interested in this technique are referred to the works of Kitamura (1980), Kitamura, et al. (1979), and Kitamura, et al. (1985).

Still another major class of FDIA schemes use the knowledge-based methodology. These knowledge-based expert systems are designed using artificial intelligence (AI) techniques. Expert systems are currently finding application to an increasing repertory of human life domains, in the center of which lies the fault diagnosis and repair domain of technological processes. Interested readers are referred to the survey paper of Pau (1986), the paper of Tzafestas (1987), and the book of Tzafestas, et al. (1989).

Some of the criteria for evaluating the performance of an FDI scheme are: a) promptness of detection, b) sensitivity to incipient faults, c) false alarm rate, d) missed fault detections, and e) incorrect fault identification. A discussion of each of these criteria is now given.

The basic function of a FDI scheme is to register an alarm when an abnormal condition develops in the system and to identify the abnormal component. Assuming that a fault is detected successfully, the issue of promptness may be of vital importance. In certain applications such as aerospace, a fault that persists for a second may destroy the mission of the operating system, if not the operating system itself.

In certain applications it may be more desirable to have reliable detection of minor faults at the sacrifice of speed in detection time or promptness. In some systems a fault detection scheme is intended to enhance maintenance operation by early detection of worn equipment. In this case promptness of detection may be of secondary importance to sensitivity. In other systems sensitivity and promptness may both be required. This leads to more complex detection schemes, possibly require both hardware and analytical redundancy.

False alarms are generally indications of poor performance of FDI schemes. Even a low false alarm rate during normal operation of the monitored plant may not be acceptable because it can quickly lead to lack of confidence in the detection scheme. However, a FDI scheme that has an acceptable false alarm rate might register a false alarm when a plant undergoes an unusual excursion, and this may be acceptable in some applications.

In other applications small faults may be so serious that it is preferable to react to false alarms, replacing unfailed components with spare parts, than to suffer deteriorated performance from an undetected, though small, fault. In these cases it is preferable to minimize the number of missed detections at the expense of the creditability of detections.

Incorrect fault identification means that the system correctly registers that a fault has occurred but incorrectly identifies the component that has failed. If the reconfiguration system proceeds to compensate for the wrong fault, it could produce a consequence as serious as a missed detection in some applications.

The compromises in detection system design among false alarm rate, sensitivity to incipient faults, and promptness of detection are difficult to make because they require extensive knowledge of the working environment and an explicit understanding of the important performance criteria of the monitored system.

In dealing with malfunctions of fault detection schemes, especially the problem of false alarms, some researchers have developed FDI schemes that use variable or adapting fault detection thresholds. Some of these techniques have demonstrated capability of reducing or minimizing false alarms. Other researchers have focused their attention on the problem of robust fault detection by design. Most of these robust fault detection schemes were designed with the goal of maximizing the sensitivity of the detector to actual sensor malfunctions, while discriminating between these faults and disturbance effects due to noise and uncertain dynamics.

Robustness of a fault detection and identification scheme can be defined as the degree to which its performance is unaffected by conditions in the operating system which turns out to be different from what they were assumed to be in the design of the scheme. Robustness problems occur with respect to four features of the operating plant: a) parameter uncertainties, b) unmodelled non-linearities or uncertain dynamics, c) disturbance and noise, and d) fault types. A brief discussion of each of these issues is now given.

Parameters refer to physical characteristics such as properties of mass, moments of inertia, electrical circuit parameters, heat transfer properties, etc. Many FDI schemes use state estimation techniques which are based on mathematical modeling of the monitored system. The models are often linearized and simplified and result in linear and time-invariant (the simplest class of dynamic systems) system representations. The inaccuracy of the model depends on the uncertainty of the values of the parameters. If all the parameters are known with precision, then state estimates can be very accurate and the FDI scheme may be remarkably sensitive to incipient faults and immune to false alarms. However, parameter values are only known approximately in most applications, especially in systems that involve fluid flows or heat transfers. Therefore, state observers or estimators have to be constructed using only nominal values for uncertain parameters. This will result in erroneous estimates. The severity of the error depends on maneuvers of the system which can not easily be determined. The algorithm or logic devices used for processing the redundant signals (e.g. state estimates) may generate false alarms, or if they are protected against this, they may fail to detect faults. This is the robust problem with respect to parameter uncertainty

Nonlinearity is a natural characteristic of all practical systems. Strictly speaking, linear dynamic systems don't exist in the real world. One of the two major reasons that we study linear systems and use linear system theory in analysis and design is that many of these nonlinear systems behave almost linearly within a narrow range of a nominal operating point. The other reason is that linear system theory is more established and easier to apply than nonlinear system theory. A FDI scheme based on linear (linearized) models could be quite satisfactory as long as the plant does not operate outside the range used for linearization. However, outside of this range nonlinearity may produce signals which are not modeled accurately by the FDI scheme. These signals may then be interpreted as faults. This is the robust problem with respect to unmodeled nonlinearity or uncertain dynamics.

Real world dynamic plants are always subject to disturbances. Disturbances are unintended system inputs originating from the operating environment. For example, wind fluctuation is a disturbance for certain systems. Disturbances are usually random signals. Furthermore, sensors are usually subject to the influence of random signals which typically originate from a different source. These random signals are called noises. Most signal processing techniques used in FDI schemes are based on the assumption that the disturbances and noises are stationary Gaussian processes and uncorrelated. If the random signals are non-stationary, non-Gaussian, or correlated in some way, then the performance of the FDI scheme will be worse than expected or even unsatisfactory. This is the robust problem with respect to disturbances and noises.

Faults can take many forms such as a nonlinearity due to wear or friction, excess noise, or a stuck value at any level within its dynamic range. Some FDI schemes are designed to detect only specific types of failures. If a malfunction or fault occurs and it is not in the repertoire, then the FDI scheme can not detect it. This is the robust problem with respect to fault types.

Some techniques have been developed by some researchers in the field of fault diagnosis to deal with some of the aforementioned robustness problems. For example, unmodeled or uncertain dynamics have been shown to act like a disturbance on a linear system in observer or state estimator based FDI schemes. The robust fault detection problem becomes one of disturbance-decoupling by design. This type of approach is known as the Unknown Input Observer Scheme (UIOS). Techniques used to deal with robustness problems with respect to fault types include hypothesis-generation and hypothesis-testing. The hypothesis-generation procedure is to build up a repertoire of known or hypothesized possible malfunctions or faults in system components or instruments (sensors). Interested readers are referred to chapter 10 and chapter 11 of the book of Patton, Frank, and Clark (1989).

The most challenging and usually missing part of research works in fault diagnosis is testing or using the FDI scheme on a real system or operating plant. Normally the application of new and developing FDI schemes to actual operating systems are prohibited because of expense or safety. If and when one does get an opportunity or authorization to test his/her FDI scheme on a real world system, numerous practical and unforeseen difficulties will present themselves. To overcome these challenges the designer of the FDI scheme must learn to understand the nature of the practical problems. This usually requires that he/she follows his/her work into a specific engineering field which may or may not be familiar to him/her. He/she has to either perform the implementation himself/herself or work very closely with the one who does the implementation. It is for this reason that most research works such as this thesis end at the simulation stage.

The large scope and great diversity of unconstrained and constrained dynamic systems prohibits a single research work to generate a general-purpose fault detection and identification approach that is applicable to all systems. In this thesis we focus our effort on a special yet major sub-class of such systems - constrained mechanical systems. The significance of studying this kind of system is threefold: a) There are many constrained mechanical systems in the real world. Some of them are used in industrial applications. b) These systems are less studied than regular (unconstrained) dynamic systems, especially in the area of fault diagnosis. c) A systematic approach to detect and identify faults in these systems has not been found but should be developed. The FDI scheme in this thesis relies purely on analytical redundancy. It is model-based and uses only quantitative reasoning. Furthermore, It falls into the sub-class of state estimator based approaches.

This thesis consists of five chapters. Chapter 2 focuses on the description, modeling, and analysis of constrained mechanical systems. It shows that constrained mechanical systems are a special class of generalized state-space systems, which are also known as descriptor systems or singular system. Some properties which are special to constrained mechanical systems are discussed in this chapter. Linearization and nonsingular transformations are performed in this chapter to yield a purely dynamic subsystem which becomes the foundation for further analysis. The result is that all subsequent analytical work can be carried out in the domain of (regular) linear system theory rather than the domain of linear singular system theory. Chapter 3 presents a design of an Unknown Input Observer (UIO) and shows how such an UIO can be used for fault detection and identification in linear or linearized dynamic systems. Similarity transformations and a nonsingular transformation are used in this chapter to help us to divide and conquer the problem. Chapter 4 uses a practical constrained mechanical system in demonstrating the applicability of the proposed unknown input observer based fault detection and identification approach. Two actuators and one sensor faults are detected and identified in the simulations of a UMS-2 robot. Finally, chapter 5 summarizes the advantages/contributions of the thesis and lists the limitations of the proposed scheme and the opportunities for further research on this subject by any interested persons in the future.

Chapter 2

Constrained Mechanical Systems

2.1 Introduction

Dynamic systems can be classified into unconstrained systems and constrained systems. Unconstrained continuous dynamic systems can be described by ordinary differential equations of motion, which are easy to simulate. All forces that do not work virtually are eliminated from the formulation of unconstrained systems. Examples of workless forces include contact forces in sliding-without-friction, rolling-without-slipping, and the internal forces maintaining rigidity of a body. On the other hand, constrained dynamic systems pose some special problems. First of all, they can no longer be described exclusively by ordinary differential equations. Presence of constraint equations makes this type of system more difficult to analyze and simulate. Additionally, because knowledge of constrained forces is crucial in some applications and such forces may not be measured directly or indirectly, estimation of constrained forces poses another issue and challenge. Let us consider a robotic manipulator (Mills & Goldenberg, 1989) performing a task on a rigid surface as an example of constrained dynamic systems. In the absence of a force sensor, the constrained forces applied by the manipulator end-effector on the environment must be estimated for control purposes so that i) neither the manipulator nor the rigid surface is damaged due to contact, ii) contact is maintained during the task, and iii) the required forces are applied to successfully complete the task. The study of constrained dynamic systems has been going on since the foundation of analytical dynamics. Understanding of analytical dynamics can be obtained from the books of Meirovitch(1970), Goldstein(1980), Greenwood(1965), Neimark(1972), and Kane & Levinson(1985). The reader is referred to the last two of the above five books for methods of deriving equations of motion for constrained dynamic systems. Basically, a constrained dynamic or mechanical system involves positions or displacements, velocities, forces, and constraints. Constraints involving only displacements or positions are called geometric constraints. Constraints involving velocities and possibly displacements as well are called velocity constraints. Geometric constraints and velocity constraints that can be integrated into geometric constraints are called holonomic constraints. Velocity constraints that can not be integrated into geometric constraints are called nonholonomic constraints. One of the major findings of past studies is that dynamic or mechanical systems with constraints result in a description of differential-algebraic equations, i.e., the natural or original representation of constrained mechanical systems in terms of a number of dynamic equations plus another number of constraint equations can be rewritten into a descriptor form (Shin and Kabamba, 1988). Descriptor systems are also called singular systems or generalized state-space systems. For information on singular systems, the readers are referred to the book of Dai(1988), the early work of Luenberger(1974 & 1978), the paper of Yip & Sincovec(1981), and the survey of Lewis(1986). The application of singular system theory to constrained mechanical systems has recently appeared as a new research topic. Generally, linear time-invariant descriptor systems can be described by the following:

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1.1}$$

$$y(t)=Cx(t) \tag{2.1.2}$$

where

$$E, A \in \Re^{n \times n}, B \in \Re^{n \times l}, C \in \Re^{m \times n}, \text{rank}E < n$$

We shall now present some definitions that will be useful in the remainder of this thesis.

Definition 2.1 - Matrix Pencil

Let E and A be two matrices of appropriate dimensions with real values. A matrix pencil is then a polynomial matrix given by (sE-A). This pencil is regular if $|sE - A| \neq 0$ for a square pencil, otherwise the pencil is singular.

Definition 2.2 - Normal Systems

Dynamic systems that can be described by only differential equations are called normal systems. An example of such a system described in state-space formulation is given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1.3}$$

$$y(t)=Cx(t) \tag{2.1.4}$$

where

$$A \in \Re^{n \times n}, B \in \Re^{n \times l}, C \in \Re^{m \times n}$$

Definition 2.3 Normal Forms for Constrained Linear Mechanical Systems

A normal form for a constrained linear mechanical system refers to the representation of the system in the form of a normal(dynamic) subsystem plus a set of algebraic constraints. For example, a normal form of the system defined in the last definition can be given by the following:

$$\dot{x}_1(t) = \overline{A}\, x_1(t) + \overline{B}\, u(t) \tag{2.1.5}$$

$$y(t) = \overline{C}\, x_1(t) \tag{2.1.6}$$

$$x_2(t) = \widetilde{A} x_1(t) + \widetilde{B}\, u(t) \tag{2.1.7}$$

where

$$\overline{A} \in \Re^{n_1 \times n_1},\ \overline{B} \in \Re^{n_1 \times l},\ \overline{C} \in \Re^{m \times n_1},\ \overline{D} \in \Re^{n_2 \times n_1}$$

$$\widetilde{A} \in \Re^{n_2 \times n_1},\ \widetilde{B} \in \Re^{n_2 \times l},\ x_1 \in \Re^{n_1},\ x_2 \in \Re^{n_2},\ n_1 + n_2 = n$$

Definition 2.4 Regularity/Solvability

A descriptor system described by equations (2.1.1) and (2.1.2) is regular or in other words has a guaranteed existence and uniqueness of a solution if and only if the following matrix pencil is regular, i.e.,

$$|sE - A| \neq 0, s \in C \tag{2.1.8}$$

Note a computationally attractive method for verifying the system's regularity is provided by Luenberger's shuffle algorithm, which can be found in the book of Dai (1989).

Definition 2.5 Infinite/Impulse Observability

A descriptor system described by equations (2.1.1) and (2.1.2) is infinitely observable or possess impulse observability if and only if

$$\text{rank}\begin{bmatrix} E & A \\ 0 & E \\ 0 & C \end{bmatrix} = n + \text{rank } E \qquad (2.1.9)$$

A more direct and more understandable definition of infinite impulse observability is as follows:

System (2.1.1)-(2.1.2) is infinite/impulse observable if the impulsive behavior of x(t) at t=0- can be uniquely determined from y(t), $t \geq 0$ in the absence of input u(t)

Definition 2.6 Finite/Reachable Observability

A descriptor system described by equations (2.1.1) and (2.1.2) is finitely observable or possess reachable observability if and only if

$$\text{rank}\begin{bmatrix} sE - A \\ C \end{bmatrix} = n \qquad \forall s \in C \qquad (2.1.10)$$

A more direct and more understandable definition of finite/reachable observability is as follows:

System (2.1.1)-(2.1.2) has finite/reachable observability if given any descriptor vector x(t), t>0 in the reachable set, it can be uniquely determined through knowledge of the output $y(\tau)$, $\tau \in (0, t]$ in the absence of input u(t).

Definition 2.7 Complete Controllability (C-controllability)

A descriptor system described by equations (2.1.1) and (2.1.2) is C-controllable if and only if

$$\text{rank}[sE\text{-}A \quad B] = n \qquad \forall s \in C \qquad (2.1.11)$$

and

$$\text{rank}[E \quad B] = n \qquad (2.1.12)$$

A more direct and more understandable definition of C-controllability is as follows:

System (2.1.1)-(2.1.2) is completely controllable (C-controllable) if there exists a control input that can make one reach any state from any initial state in a finite time period.

Definition 2.8 Reachable Controllability (R-controllability)

A descriptor system described by equations (2.1.1) and (2.1.2) is R-controllable if and only if

$$\text{rank}[sE-A \quad B] = n \qquad \forall s \in C \qquad (2.1.13)$$

A more direct and more understandable definition of R-controllability is as follows:

System (2.1.1)-(2.1.2) is R-controllable if there exists an admissible control that can make the state of the system to go from any initial state to a point in the set of reachable states (a subspace of $\Re^n$).

The above definitions are of value and will be used in the rest of the thesis. In section 2.2, a natural mathematical description of constrained non-linear mechanical systems is initially given in the form of dynamic equations plus constraint equations. The non-linear representation is then linearized. In section 2.3, any possible redundancy in the constraints is eliminated and the linearized mathematical model is rewritten into a special form as well as a descriptor form. The special form is needed for deriving a normal form of the representation. In section 2.4, a normal form of the linear mechanical descriptor system is derived. In section 2.5 and 2.6, properties of linear mechanical descriptor systems and their impacts on observer design for such systems are discussed. Finally, section 2.7 summarizes this chapter and explains the link between this chapter and subsequent chapters.

2.2 Description of Constrained Dynamic Systems

Constraints in dynamic systems can be classified as scleronomic constraints or rheonomic constraints depending on whether the time variable t is explicitly contained in the constraints. Systems with time-invariant constraints are called scleronomic systems. Systems with time-varying constraints are called rheonomic systems. The most common model for dynamic systems with constraints is that of Lagrange's equations. Modeling of constrained dynamic systems using Lagrange's equations can be found in the book of Goldstein (1980). According to Shin and Kabamba (1988), Constrained dynamic systems can be modeled as:

$$M(q)\ddot{q} + H(q,\dot{q}) = J^T(q,\dot{q})\,\Lambda + T \qquad (2.2.1)$$

$$\theta(q) = 0 \qquad (2.2.2)$$

$$\varphi(q,\dot{q}) = 0 \tag{2.2.3}$$

$$r = \psi(q,\dot{q}) \tag{2.2.4}$$

where

$q(t) \in \Re^n$ is the generalized coordinates vector

$\dot{q}(t) \in \Re^n$ is the velocity vector

$\ddot{q}(t) \in \Re^n$ is the accelerations vector

$M(q) \in \Re^{n \times n}$ is the symmetric positive definite inertia matrix

$H(q,\dot{q}) \in \Re^n$ is the force vector

θ : represents a set of holonomic constraints

φ : represents a set of nonholonomic constraints

$\Lambda \in \Re^p$ is the Lagrange multiplier vector

$T \in \Re^n$ represents input forces acting as controls

$r \in \Re^m$ is an output vector

and

$J^T(q,\dot{q})$ is called the Jacobian of constraint equations which is defined as

(a) In the case of only holonomic constraints represented by (2.2.2)

$$J^T(q) = \frac{\partial \theta(q)}{\partial q}$$

(b) In the case of only nonholonomic constraints represented by (2.2.3)

$$J^T(q,\dot{q}) = \frac{\partial \varphi(q,\dot{q})}{\partial \dot{q}}$$

(c) In the case of both holonomic and nonholonomic constraints represented by (2.2.2) and (2.2.3)

$$J^T(q,\dot{q}) = \begin{bmatrix} \dfrac{\partial \varphi(q,\dot{q})}{\partial \dot{q}} \\ \dfrac{\partial \theta(q)}{\partial q} \end{bmatrix}$$

Note that since $q(t)$ represents generalized coordinates, its components are independent and the constraint equations in (2.2.2) and (2.2.3) are linearly independent.

The process of linearizing the system represented by (2.2.1-(2.2.4) requires multivariable Taylor series expansions involving only the first order terms. Given a nominal state $(q_0, \dot{q}_0, \ddot{q}_0, \Lambda_0, T_0)$, let us first define the following notations:

$$
\begin{aligned}
z &= q - q_0 \\
\dot{z} &= \dot{q} - \dot{q}_0 \\
\ddot{z} &= \ddot{q} - \ddot{q}_0 \\
\lambda &= \Lambda - \Lambda_0 \\
f &= T - T_0 \\
y &= r - r_0
\end{aligned}
$$

Then we have the following first order Taylor series expansions.

$$M(q) = M(q_0) + \frac{\partial M(q)}{\partial q}\Big|_{(q_0)} z \tag{2.2.5}$$

$$\ddot{q} = \ddot{q}_0 + \ddot{z} \tag{2.2.6}$$

$$H(q,\dot{q}) = H(q_0,\dot{q}_0) + \frac{\partial H(q,\dot{q})}{\partial q}\Big|_{(q_0,\dot{q}_0)} z + \frac{\partial H(q,\dot{q})}{\partial \dot{q}}\Big|_{(q_0,\dot{q}_0)} \dot{z} \tag{2.2.7}$$

$$\Lambda = \Lambda_0 + \lambda \tag{2.2.8}$$

$$T = T_0 + f \tag{2.2.9}$$

$$\theta(q) = \theta(q_0) + \frac{\partial \theta(q)}{\partial q}\Big|_{(q_0)} z \tag{2.2.10}$$

$$\varphi(q,\dot{q}) = \varphi(q_0,\dot{q}_0) + \frac{\partial \varphi(q,\dot{q})}{\partial q}\Big|_{(q_0,\dot{q}_0)} z + \frac{\partial \varphi(q,\dot{q})}{\partial \dot{q}}\Big|_{(q_0,\dot{q}_0)} \dot{z} \tag{2.2.11}$$

$$\psi(q,\dot{q}) = \psi(q_0,\dot{q}_0) + \frac{\partial \psi(q,\dot{q})}{\partial q}\Big|_{(q_0,\dot{q}_0)} z + \frac{\partial \psi(q,\dot{q})}{\partial \dot{q}}\Big|_{(q_0,\dot{q}_0)} \dot{z} \tag{2.2.12}$$

and the Jacobian takes one of the following forms depending on the types of constraints:

$$J^T(q) = J^T(q_0) + \left(\frac{\partial J(q)}{\partial q}\Big|_{q_0, \dot q_0} z \right)^T \qquad (2.2.13)$$

or

$$J^T(q,\dot q) = J^T(q_0, \dot q_0) + \left(\frac{\partial J(q,\dot q)}{\partial q}\Big|_{q_0, \dot q_0} z \right)^T + \left(\frac{\partial J(q,\dot q)}{\partial \dot q}\Big|_{q_0, \dot q_0} \dot z \right)^T \qquad (2.2.14)$$

Substituting (2.2.5)-(2.2.11) into (2.2.1)-(2.2.4) and simplifying the resultant equations results in:

$$M\ddot z + D\dot z + K z = f + J^T \lambda \qquad (2.2.15)$$

$$L z = 0 \qquad (2.2.16)$$

$$G \dot z + H z = 0 \qquad (2.2.17)$$

$$y = C_p z + C_v \dot z \qquad (2.2.18)$$

where

$$M = M(q_0)$$

$$D = \frac{\partial H(q,\dot q)}{\partial \dot q}\Big|_{q_0, \dot q_0}$$

$$K = \frac{\partial H(q,\dot q)}{\partial q}\Big|_{q_0, \dot q_0} + \frac{\partial M(q)}{\partial q}\Big|_{q_0, \dot q_0} \ddot q_0 - \Lambda_0 \frac{\partial J^T(q)}{\partial q}\Big|_{q_0, \dot q_0}$$

$$L = \frac{\partial \theta(q)}{\partial q}\Big|_{q_0, \dot q_0}$$

$$G = \frac{\partial \varphi(q,\dot q)}{\partial \dot q}\Big|_{q_0, \dot q_0}$$

$$H = \frac{\partial \varphi(q,\dot q)}{\partial q}\Big|_{q_0, \dot q_0}$$

$$C_p = \frac{\partial \psi^T(q,\dot q)}{\partial q}\Big|_{q_0, \dot q_0}$$

$$C_v = \frac{\partial \psi(q,\dot q)}{\partial \dot q}\Big|_{q_0, \dot q_0}$$

The holonomic constraints in equation (2.2.16) and the nonholonomic constraints in equation (2.2.17) can be represented in the following generalized form:

$$\overline{G}\ \dot{z} + \overline{H}\ z = 0 \tag{2.2.19}$$

where

$\overline{G} = 0$ and $\overline{H} = L$ in the case of only holonomic constraint

$\overline{G} = G$ and $\overline{H} = H$ in the case of only nonholonomic constraint

$\overline{G} = \begin{bmatrix} G \\ 0 \end{bmatrix}$ and $\overline{H} = \begin{bmatrix} H \\ L \end{bmatrix}$ in the case of both types of constraints

Therefore linear or linearized constrained mechanical systems have the following form:

$$M\ddot{z} + D\dot{z} + Kz = f + J^T\lambda \tag{2.2.20}$$

$$\overline{G}\ \dot{z} + \overline{H}\ z = 0 \tag{2.2.21}$$

$$y = C_p\ z + C_v\ \dot{z} \tag{2.2.22}$$

The above representation can be rewritten in a linear descriptor form:

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \overline{H} & \overline{G} & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} \tag{2.2.23}$$

$$y = \begin{bmatrix} C_p & C_v & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} \tag{2.2.24}$$

2.3 Special Form of Constrained Linear Mechanical Systems

In this section we will perform a nonsingular transformation on the generalized constraint equation (2.2.19). The motivation of this transformation is best understood in the next section(2.4). The process of this transformation yields a nonsingular(orthogonal) transformation matrix T and a special form of the linear descriptor system representation. Both will be used in deriving the normal form of the descriptor system in the section 2.4.

We start with a matrix pencil ($\lambda \bar{G} + \bar{H}$)

where

λ is a complex variable in the complex plane or Laplace operator

$\bar{G}$, $\bar{H} \in \Re^{q \times n}$

q is the number of holonomic plus nonholonomic constraints

First, let us define row compression matrix and column compression matrix for an arbitrary singular matrix denoted by H. According to singular value decomposition theory in linear algebra, Orthogonal matrices R and Q exist such that

$$R^T H Q = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \quad \Leftrightarrow \quad H = R \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} Q^T$$

where

Σ is a diagonal matrix filled with singular values of H

Then the following equations can be established

$$R^T H = R^T R \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} Q^T = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = \begin{bmatrix} \Sigma Q_1^T \\ 0 \end{bmatrix}$$

$$H Q = R \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} Q^T Q = \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_1 \Sigma & 0 \end{bmatrix}$$

Thus R^T can be used as a row compression matrix and Q can be used as a column compression matrix

Now perform the row compression of $\overline{G}$ using an orthogonal matrix P_1 such that

$$P_1 \overline{G} = \begin{bmatrix} \widetilde{G}_1 \\ 0 \end{bmatrix} \text{and} \quad P_1 \overline{H} = \begin{bmatrix} \widetilde{H}_1 \\ \widetilde{H}_2 \end{bmatrix}$$

then, we have

$$P_1 (\lambda \overline{G} + \overline{H}) = \begin{bmatrix} \lambda \widetilde{G}_1 + \widetilde{H}_1 \\ \widetilde{H}_2 \end{bmatrix}$$

Perform further column and row compression of $\widetilde{H}_2$ using orthogonal matrices P_2 and T_1 such that

$$P_2 \widetilde{H}_2 T_1 = \begin{bmatrix} 0 & H_1 \\ 0 & 0 \end{bmatrix}$$

where

H_1 is a nonsingular matrix

Thus, we have

$$\begin{bmatrix} I & 0 \\ 0 & P_2 \end{bmatrix} P_1 (\lambda \overline{G} + \overline{H}) T_1 = \begin{bmatrix} \lambda \overline{G}_1 + \overline{H}_1 & \times \\ 0 & \hat{H}_1 \end{bmatrix}, \qquad \hat{H}_1 = \begin{bmatrix} H_1 \\ 0 \end{bmatrix}$$

where ' × ' indicates a usually nonzero matrix pencil. Then, perform the same operations on subpencil $\lambda \overline{G}_1 + \overline{H}$ as on $\lambda \overline{G} + \overline{H}$. Repeat the process until $\overline{G}_k$ in the resulting subpencil $\lambda \overline{G}_k + \overline{H}_k$ is of full row rank. Hence, we have

$$\begin{bmatrix} I & 0 \\ 0 & P_{k+1} \end{bmatrix} P_k \cdots \begin{bmatrix} I & 0 \\ 0 & P_2 \end{bmatrix} P_1 (\lambda \overline{G} + \overline{H}) T_1 \ldots T_k = \begin{bmatrix} \lambda G_k + H_k & \times & \cdots & \cdots & \cdots & \times \\ 0 & \hat{H}_k & \times & & & \vdots \\ 0 & 0 & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \times & \vdots \\ 0 & 0 & 0 & 0 & \ddots & \times \\ 0 & 0 & 0 & 0 & 0 & \hat{H}_1 \end{bmatrix},$$

Finally, perform the column compression of $\overline{G}_k$ to get

$$(\lambda \overline{G}_k + \overline{H}_k) T_k = [\widetilde{H}_{k1} \ \ \lambda \widetilde{G}_k + \widetilde{H}_{k2}]$$

where

$\overline{G}^{*}_{k}$ is nonsingular

and therefore the above equation can be rewritten as

$$P(\lambda \overline{G}+\overline{H})T = \begin{bmatrix} \overline{H}_{k,1} & \lambda I+\overline{H}_{k,2} & \times & \cdots & \cdots & \cdots & \times \\ 0 & 0 & \hat{I}_{k} & \times & & & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ddots & \times & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ddots & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{I}_{1} \end{bmatrix} \quad (2.3.1)$$

where

$$\hat{I}_{1} = \begin{bmatrix} I_{n_1} \\ 0 \end{bmatrix}$$

$$P = \begin{bmatrix} I & 0 \\ 0 & P_{k-1} \end{bmatrix} P_{k} \cdots \begin{bmatrix} I & 0 \\ 0 & P_{2} \end{bmatrix} P_{1} \quad \text{is nonsingular} \quad (2.3.2)$$

and

$$T = T_{1} \ldots T_{k} \qquad \text{is orthogonal} \quad (2.3.3)$$

Now let us take Laplace transform on the constraints equation $\overline{G}\ \dot{z} + \overline{H}\ z = 0$

$$(\lambda \overline{G}+\overline{H})\, Z(\lambda) = 0 \quad (2.3.4)$$

Let us further define a new generalized state vector

$$\xi = T^{-1} z = T^{T} z \quad (2.3.5)$$

then

$$z = T\ \xi \quad (2.3.6)$$

Taking Laplace transform on (2.3.6) yields

$$Z(\lambda) = T\ \Xi(\lambda) \quad (2.3.7)$$

Premultiply equation (2.3.4) by P and substituting (2.3.7) into it results

$$P\,(\lambda \overline{G}+\overline{H})\,T\ \Xi(\lambda) = 0 \quad (2.3.8)$$

Substituting (2.3.1) into (2.3.8) and partitioning $\Xi(\lambda)$ results in

$$\begin{bmatrix} \overline{H}_{k,1} & \lambda I + \overline{H}_{k,2} & * & \cdots & \cdots & \cdots & * \\ 0 & 0 & \hat{I}_k & * & & & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ddots & * & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ddots & * \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{I}_1 \end{bmatrix} \begin{bmatrix} \Xi_1(\lambda) \\ \Xi_2(\lambda) \\ \Xi_3(\lambda) \end{bmatrix} = 0 \tag{2.3.9}$$

Equation (2.3.9) further results in the following two equations:

$$\overline{H}_{k,1}\,\Xi_1(\lambda) + (\lambda I + \overline{H}_{k,2})\,\Xi_2(\lambda) + \begin{bmatrix} * & \cdots & \cdots & * \end{bmatrix} \Xi_3(\lambda) = 0 \tag{2.3.10}$$

$$\begin{bmatrix} \hat{I}_k & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \hat{I}_1 \end{bmatrix} \Xi_3(\lambda) = 0 \tag{2.3.11}$$

Simplifying the above two equations yields

$$\overline{H}_{k,1}\,\Xi_1(\lambda) + (\lambda I + \overline{H}_{k,2})\,\Xi_2(\lambda) = 0 \tag{2.3.12}$$

Taking inverse Laplace transformation of (2.3.12) results in

$$\overline{H}_{k,1}\,\xi_1(t) + \dot{\xi}_2(t) + \overline{H}_{k,2}\,\xi_2(t) = 0 \tag{2.3.13}$$

Equation (2.3.13) can be rewriten as

$$\begin{bmatrix} 0 & I & 0 \end{bmatrix} \begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \\ \dot{\xi}_3(t) \end{bmatrix} + \begin{bmatrix} \overline{H}_{k,1} & \overline{H}_{k,2} & 0 \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \end{bmatrix} = 0 \tag{2.3.14}$$

Substituting (2.3.5) into (2.3.14) yields

$$\begin{bmatrix} 0 & I & 0 \end{bmatrix} T^{-1}\,\dot{z}(t) + \begin{bmatrix} \overline{H}_{k,1} & \overline{H}_{k,2} & 0 \end{bmatrix} T^{-1}\,z(t) = 0 \tag{2.3.15}$$

Define

$$N = \begin{bmatrix} 0 & I & 0 \end{bmatrix} T^{-1} \qquad (2.3.16)$$

and

$$S = \begin{bmatrix} \overline{H}_{k1} & \overline{H}_{k2} & 0 \end{bmatrix} T^{-1} = \begin{bmatrix} S_1 & S_2 & 0 \end{bmatrix} T^{-1} \qquad (2.3.17)$$

Then we obtain the following results

$$N\, \dot{z}(t) + S\, z(t) = 0 \qquad (2.3.18)$$

$$N\, T = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \qquad (2.3.19)$$

$$S\, T = \begin{bmatrix} S_1 & S_2 & 0 \end{bmatrix} \qquad (2.3.20)$$

The above results can be summarized and stated as the following theorem:

Theorem 2.3.1 - Through the nonsingular transformation of matrix pencils, constraint equations (2.2.16) and (2.2.17) can be transformed into one of the following equivalent forms:

(a) holonomic constraints

$$F\, z = 0, \qquad F \in \Re^{q \times n} \qquad (2.3.21)$$

(b) nonholonomic constraints

$$N \dot{z} + S\, z = 0, \quad N, S \in \Re^{q \times n} \qquad (2.3.22)$$

(c) holonomic and nonholonomic constraints

$$N \dot{z} + S\, z = 0 \quad \text{and} \quad F\, z = 0 \qquad (2.3.23)$$

where

$$N, S \in \Re^{q_s \times n}, \qquad F \in \Re^{q_p \times n}, \qquad q_s + q_p = q$$

q is the number of independent constraints

q_p is the number of independent holonomic constraints

q_s is the number of independent nonholonomic constraints

The Jacobian J will be one of the following forms:

(a) For constraint equation (2.2.7) $\qquad J=F \qquad$ (2.3.24)

(b) For constraint equation (2.2.8) $\qquad J=N \qquad$ (2.3.25)

(c) For constraint equation (2.2.15) $$J = \begin{bmatrix} N \\ F \end{bmatrix} \quad (2.3.26)$$

Moreover, from the transformation which brings equation (2.2.7) and (2.2.8) into one of its special forms (2.3.21), (2.3.22) or (2.3.23), an orthogonal matrix T, i.e., $T^{-1}=T^T$, can be obtained such that

(a) for constraint equation (2.2.7): $$FT = \begin{bmatrix} 0 & I_{q} \end{bmatrix} \quad (2.3.27)$$

(b) for constraint equation (2.2.8): $$NT= \begin{bmatrix} 0 & I_{q} \end{bmatrix},\ ST= \begin{bmatrix} S_1 & S_2 \end{bmatrix} \quad (2.3.28)$$

(c) for constraint equation (2.2.15):

$$NT= \begin{bmatrix} 0 & I_{q_s} & 0 \end{bmatrix},\ ST=\begin{bmatrix} S_1 & S_2 & 0 \end{bmatrix},\ FT= \begin{bmatrix} 0 & 0 & I_{q_F} \end{bmatrix} \quad (2.3.29)$$

The above constraints can be denoted in a generalized form

$$\bar{N}\dot{z} + \bar{S}z = 0 \quad (2.3.30)$$

where

$\bar{N} = 0$, $\bar{S}$ = F in the case of only holonomic constraints

$\bar{N}$ = N, $\bar{S}$ = S in the case of only nonholonomic constraints

$\bar{N} = \begin{bmatrix} N \\ 0 \end{bmatrix}$, $\bar{S} = \begin{bmatrix} S \\ F \end{bmatrix}$ in the case of both kinds of constraints

Thus, the special form representation of linear mechanical system can be written as

$$\begin{bmatrix} I & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \bar{S} & \bar{N} & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} \quad (2.3.32)$$

$$y = \begin{bmatrix} C_p & C_v & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} \quad (2.3.33)$$

The above results will be used in deriving a dynamic subsystem(normal form) for the linear mechanical system in the following section.

Example

We now use an example to illustrate how the special form transformation is performed, i.e., we will apply theorem 2.3.1 to a specific system. The example used here is a rolling ring drive which has one holonomic constraint and one non-holonomic constraint. This system was found in the paper, of Hou et.al. (1993). The linearized system representation has the following form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ddot{z} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \dot{z} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} z = \begin{bmatrix} u \\ 0 \\ 0 \\ 0 \end{bmatrix} + J^T \lambda \qquad (2.3.34)$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \dot{z} - \begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} z = 0 \qquad (2.3.35)$$

$$\begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix} z = 0 \qquad (2.3.36)$$

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} z + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \dot{z} \qquad (2.3.37)$$

which corresponds to the form in equations (2.2.15) - (2.2.18).

The matrix pencil ($\lambda \overline{G} + \overline{H}$) for this particular system would be

$$\lambda \overline{G} + \overline{H} = \lambda \begin{bmatrix} G \\ 0 \end{bmatrix} + \begin{bmatrix} H \\ L \end{bmatrix} = \begin{bmatrix} \lambda G & H \\ 0 & L \end{bmatrix} = \lambda \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \qquad (2.3.38)$$

$$\Rightarrow \qquad \lambda \overline{G} + \overline{H} = \begin{bmatrix} -1 & 0 & 0 & \lambda + 1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \qquad (2.3.39)$$

Since the above matrix pencil is already in row compressed form, there is no need to perform row compression Therefore we have the following

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (2.3.40)$$

$$\widetilde{G}_1 = G \tag{2.3.41}$$

$$\widetilde{H}_1 = H \tag{2.3.42}$$

$$P_1(\lambda\overline{G} + \overline{H}) = \begin{bmatrix} \lambda\widetilde{G}_1 + \widetilde{H}_1 \\ \widetilde{H}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & \lambda+1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \tag{2.3.43}$$

The last equation (2.3.43) means that

$$\widetilde{H}_2 = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix} \tag{2.3.44}$$

Now we need to find orthogonal matrices P_2 and T_1 such that

$$P_2\,\widetilde{H}_2 T_1 = \begin{bmatrix} 0 & 0 & 0 & H_1 \end{bmatrix} \tag{2.3.45}$$

It can be verified that the following orthogonal matrices satisfy (2.3.45)

$$P_2 = 1 \tag{2.3.46}$$

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.3.47}$$

$$1\begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \sqrt{2} \end{bmatrix} \tag{2.3.48}$$

Then, we have

$$\begin{bmatrix} I & 0 \\ 0 & P_2 \end{bmatrix} P_1(\lambda\overline{G} + \overline{H})\,T_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 0 & \lambda+1 \\ 0 & 1 & -1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & \lambda+1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}$$

$$= \begin{bmatrix} \overline{H}_{1,1} & \lambda+\overline{H}_{1,2} & 0 \\ 0 & 0 & \hat{I}_1 \end{bmatrix}$$

Therefore we have

$$\overline{H}_{1,1} = \begin{bmatrix} -1 & 0 \end{bmatrix} \tag{2.3.49}$$

$$\overline{H}_{1,2} = 1 \tag{2.3.50}$$

The transformation matrix T is determined as

$$\mathrm{T} = \mathrm{T}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.3.51}$$

Then we can calculate the following matrices:

$$\mathrm{S} = \begin{bmatrix} \overline{H}_{1,1} & \overline{H}_{1,2} & 0 \end{bmatrix} \mathrm{T}^{-1} = \begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1}$$

$$\Rightarrow \qquad \mathrm{S} = \begin{bmatrix} -1 & 0 & 0 & 1 \end{bmatrix} \tag{2.3.52}$$

$$\mathrm{N} = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \mathrm{T}^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1}$$

$$\Rightarrow \qquad N = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3.53}$$

$$F = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} T^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1}$$

$$\Rightarrow \qquad F = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} \tag{2.3.54}$$

$$\bar{N} = \begin{bmatrix} N \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.3.55}$$

$$\bar{S} = \begin{bmatrix} S \\ F \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} \tag{2.3.56}$$

$$J = \begin{bmatrix} N \\ F \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} \tag{2.3.57}$$

Finally we obtain the special form representation of the linearized system as follows:

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \bar{S} & \bar{N} & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} \tag{2.3.58}$$

$$y = \begin{bmatrix} C_p & C_v & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} \tag{2.3.59}$$

The numerical representation of equations (2.3.58) and (2.3.59) is given in the following

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \\ \ddot{z}_4 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2
\end{bmatrix}
=
$$

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & \frac{1}{\sqrt{2}} \\
0 & 1 & -2 & 1 & 0 & 1 & -2 & 1 & 0 & -\frac{1}{\sqrt{2}} \\
0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & 0 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\ z_2 \\ z_3 \\ z_4 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \lambda_1 \\ \lambda_2
\end{bmatrix}
+
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ u \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
\qquad (2.3.60)
$$

$$
y =
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
z_1 \\ z_2 \\ z_3 \\ z_4 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \lambda_1 \\ \lambda_2
\end{bmatrix}
\qquad (2.3.61)
$$

This concludes our illustration of transforming a linear descriptor system to its special form.

2.4 Normal Forms for Constrained Linear Mechanical Systems

In this section we will derive normal forms for linear mechanical systems with various constraints. First let us consider the most general case which involves both holonomic and nonholomonic constraints. Let n be the number of descriptor variables, q be the total number of independent holonomic and nonholonomic constraints, q_v be the number of independent nonholonomic constraints, and q_p be the number of independent holonomic constraints. Using the orthogonal transformation matrix T ($T\,T^T = T^T T = I$) developed in the last section enables us to do the following:

Partition the transformed generalized state vector defined in equation (2.3.5) as

$$T^T z = \xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \qquad (2.4.1)$$

where

$$\xi_1 \in \Re^{n-q}, \quad \xi_2 \in \Re^{q_v}, \quad \xi_3 \in \Re^{q_p}$$

Partition the transformed input vector as

$$T^T f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \qquad (2.4.2)$$

where

$$f_1 \in \Re^{n-q}, \quad f_2 \in \Re^{q_v}, \quad f_3 \in \Re^{q_p}$$

Partition the transformed mass matrix M, the stiffness matrix K, and the damping matrix D as

$$T^T M T = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \qquad (2.4.3)$$

$$T^T KT = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \tag{2.4.4}$$

$$T^T DT = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \tag{2.4.5}$$

Let Λ denotes M, K, or D and the dimensions of the above partitions be as follows:

$$\Lambda_{11} \in \Re^{(n-q)\times(n-q)} \qquad \Lambda_{12} \in \Re^{(n-q)\times q_s} \qquad \Lambda_{12} \in \Re^{(n-q)\times q_p}$$

$$\Lambda_{21} \in \Re^{q_s \times (n-q)} \qquad \Lambda_{22} \in \Re^{q_s \times q_s} \qquad \Lambda_{23} \in \Re^{q_s \times q_p}$$

$$\Lambda_{31} \in \Re^{q_p \times (n-q)} \qquad \Lambda_{32} \in \Re^{q_p \times q_p} \qquad \Lambda_{33} \in \Re^{q_p \times q_p}$$

Pre-multiplying both sides of the descriptor form representation of the last section by a nonsingular matrix $Q = \mathrm{diag}(T^T, T^T, I_q)$ and noting that $z = T\xi$ results in

$$\begin{bmatrix} T^T & 0 & 0 \\ 0 & T^T & 0 \\ 0 & 0 & I_q \end{bmatrix}\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} T\dot{\xi} \\ T\ddot{\xi} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} T^T & 0 & 0 \\ 0 & T^T & 0 \\ 0 & 0 & I_q \end{bmatrix}\begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \overline{S} & \overline{N} & 0 \end{bmatrix}\begin{bmatrix} T\xi \\ T\dot{\xi} \\ \lambda \end{bmatrix}$$

$$+\begin{bmatrix} T^T & 0 & 0 \\ 0 & T^T & 0 \\ 0 & 0 & I_q \end{bmatrix}\begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} \tag{2.4.6}$$

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & T^T MT & 0 \\ 0 & 0 & 0_q \end{bmatrix}\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0_n & I_n & 0 \\ -T^T KT & -T^T DT & T^T J^T \\ \overline{S}T & \overline{N}T & 0 \end{bmatrix}\begin{bmatrix} \xi \\ \dot{\xi} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ T^T f \\ 0 \end{bmatrix} \tag{2.4.7}$$

Substituting (2.4.1) through (2.4.5) and the following results from the last section

$$\bar{S}=\begin{bmatrix} S \\ F \end{bmatrix},\ \bar{N}=\begin{bmatrix} N \\ 0 \end{bmatrix},\ \mathrm{J}=\begin{bmatrix} N \\ F \end{bmatrix},\ \mathrm{NT}=\begin{bmatrix} 0 & I_{q_v} & 0 \end{bmatrix},\ \mathrm{ST}=\begin{bmatrix} S_1 & S_2 & 0 \end{bmatrix},\ \mathrm{FT}=\begin{bmatrix} 0 & 0 & I_{q_p} \end{bmatrix}$$

into (2.4.7) results in

$$\begin{bmatrix} I_{n-q} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{q_v} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{q_p} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{11} & M_{12} & M_{13} & 0 & 0 \\ 0 & 0 & 0 & M_{21} & M_{22} & M_{23} & 0 & 0 \\ 0 & 0 & 0 & M_{31} & M_{32} & M_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \ddot{\xi}_1 \\ \ddot{\xi}_2 \\ \ddot{\xi}_3 \\ \dot{\lambda}_v \\ \dot{\lambda}_p \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & I_{n-q} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{q_v} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{q_p} & 0 & 0 \\ -K_{11} & -K_{12} & -K_{13} & -D_{11} & -D_{12} & -D_{13} & 0 & 0 \\ -K_{21} & -K_{22} & -K_{23} & -D_{21} & -D_{22} & -D_{23} & I_{q_v} & 0 \\ -K_{31} & -K_{32} & -K_{33} & -D_{31} & -D_{32} & -D_{33} & 0 & I_{q_p} \\ S_1 & S_2 & 0 & 0 & I_{q_v} & 0 & 0 & 0 \\ 0 & 0 & I_{q_p} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \lambda_v \\ \lambda_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ f_1 \\ f_2 \\ f_3 \\ 0 \\ 0 \end{bmatrix} \tag{2.4.8}$$

where

$$\lambda = \begin{bmatrix} \lambda_v \\ \lambda_p \end{bmatrix}$$

Note that (2.4.8) can be expanded into as many equations as its number of rows. The seventh and eighth rows of (2.4.8) offer the following equations:

$$0 = S_1\xi_1 + S_2\xi_2 + \dot{\xi}_2 \tag{2.4.9}$$

$$0 = \xi_3 \tag{2.4.10}$$

These two equations result in the following:

$$\dot{\xi}_2 = -S_1\xi_1 - S_2\xi_2 \tag{2.4.11}$$

$$\ddot{\xi}_2 = -S_1\dot{\xi}_1 - S_2\dot{\xi}_2 \tag{2.4.12}$$

$$\xi_3 = 0 \tag{2.4.13}$$

$$\dot{\xi}_3 = 0 \tag{2.4.14}$$

$$\ddot{\xi}_3 = 0 \tag{2.4.15}$$

The fourth row of (2.4.8) offers the following equation:

$$M_{11}\ddot{\xi}_1 + M_{12}\ddot{\xi}_2 + M_{13}\ddot{\xi}_3 = -K_{11}\xi_1 - K_{12}\xi_2 - K_{13}\xi_3$$
$$- D_{11}\dot{\xi}_1 - D_{12}\dot{\xi}_2 - D_{13}\dot{\xi}_3 \tag{2.4.16}$$

Substituting (2.4.11)-(2.4.15) into (2.4.16) and rearranging terms results in the following:

$$-M_{12}S_1\dot{\xi}_1 - M_{12}S_2\dot{\xi}_2 + M_{11}\ddot{\xi}_1 = (-K_{11} + D_{12}S_1)\xi_1$$
$$+ (-K_{12} + D_{12}S_2)\xi_2 - D_{11}\dot{\xi}_1 \tag{2.4.17}$$

The fifth and sixth rows of (2.4.8) offer the following equations:

$$M_{21}\ddot{\xi}_1 + M_{22}\ddot{\xi}_2 + M_{23}\ddot{\xi}_3 = -K_{21}\xi_1 - K_{22}\xi_2 - K_{23}\xi_3$$
$$- D_{21}\dot{\xi}_1 - D_{22}\dot{\xi}_2 - D_{23}\dot{\xi}_3 + \lambda_p + f_2 \tag{2.4.18}$$

and

$$M_{31}\ddot{\xi}_1 + M_{32}\ddot{\xi}_2 + M_{33}\ddot{\xi}_3 = -K_{31}\xi_1 - K_{32}\xi_2 - K_{33}\xi_3$$
$$- D_{31}\dot{\xi}_1 - D_{32}\dot{\xi}_2 - D_{33}\dot{\xi}_3 + \lambda_p + f_3 \tag{2.4.19}$$

Substituting (2.4.12)- (2.4.15) into (2.4.18)-(2.4.19) and writing the two equations in a matrix form result in the following

$$\begin{bmatrix} M_{21} \\ M_{31} \end{bmatrix} \ddot{\xi}_1 + \begin{bmatrix} M_{22} \\ M_{32} \end{bmatrix} \ddot{\xi}_2 = -\begin{bmatrix} K_{21} \\ K_{31} \end{bmatrix} \xi_1 - \begin{bmatrix} K_{22} \\ K_{32} \end{bmatrix} \xi_2 - \begin{bmatrix} D_{21} \\ D_{31} \end{bmatrix} \dot{\xi}_1 - \begin{bmatrix} D_{22} \\ D_{32} \end{bmatrix} \dot{\xi}_2 - \begin{bmatrix} \lambda_r \\ \lambda_p \end{bmatrix} + \begin{bmatrix} f_2 \\ f_3 \end{bmatrix} \quad (2.4.20)$$

Substituting (2.4.11) into (2.4.20) and re-arranging terms result in the following equation:

$$-\overline{M}_{22} S_1 \dot{\xi}_1 - \overline{M}_{22} S_2 \dot{\xi}_2 + \overline{M}_{21} \ddot{\xi}_1 = (-\overline{K}_{21} + \overline{D}_{21} S_1)\, \xi_1 + (-\overline{K}_{22} + \overline{D}_{22} S_2)\, \xi_2 - \overline{D}_{21} \dot{\xi}_1 + \lambda + \bar{f}_2 \quad (2.4.21)$$

where

$$\overline{M}_{21} = \begin{bmatrix} M_{21} \\ M_{31} \end{bmatrix}, \overline{M}_{22} = \begin{bmatrix} M_{22} \\ M_{32} \end{bmatrix}, \overline{K}_{21} = \begin{bmatrix} K_{21} \\ K_{31} \end{bmatrix}, \overline{K}_{22} = \begin{bmatrix} K_{22} \\ K_{32} \end{bmatrix}, \overline{D}_{21} = \begin{bmatrix} D_{21} \\ D_{31} \end{bmatrix}, \overline{D}_{22} = \begin{bmatrix} D_{22} \\ D_{32} \end{bmatrix}$$

and

$$\bar{f}_2 = \begin{bmatrix} f_2 \\ f_3 \end{bmatrix}$$

In view of equations (2.4.11), (2.4.17), and (2.4.21), equation (2.4.8) can be rewritten as

$$\begin{bmatrix} I_{n-q} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{q_s} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -M_{12}S_1 & -M_{12}S_2 & 0 & M_{11} & 0 & 0 & 0 \\ -\overline{M}_{22}S_1 & -\overline{M}_{22}S_2 & 0 & M_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \ddot{\xi}_1 \\ \ddot{\xi}_2 \\ \ddot{\xi}_3 \\ \dot{\lambda} \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & I_{n-q} & 0 & 0 & 0 \\ -S & -S_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{q_r} & 0 \\ -K_{11} + D_{12}S_1 & -K_{12} + D_{12}S_2 & 0 & -D_{11} & 0 & 0 & 0 \\ -\overline{K}_{21} + \overline{D}_{21}S_1 & -\overline{K}_{22} + \overline{D}_{22}S_2 & 0 & -\overline{D}_{21} & 0 & 0 & I_q \\ S_1 & S_2 & 0 & 0 & I_{q_s} & 0 & 0 \\ 0 & 0 & 0 & I_{q_p} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ f_1 \\ \bar{f}_2 \\ 0 \\ 0 \end{bmatrix} \quad (2.4.22)$$

Equation (2.4.22) can be rewritten in the following form by re-arranging or re-grouping variables in the generalized state vector:

$$
\begin{bmatrix}
I_{n-q} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & I_{q_s} & 0 & 0 & 0 & 0 & 0 \\
-M_{12}S_1 & -M_{12}S_2 & M_{11} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\overline{M}_{22}S_1 & -\overline{M}_{22}S_2 & \overline{M}_{21} & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_1 \\ \dot{\xi}_2 \\ \ddot{\xi}_1 \\ \dot{\xi}_3 \\ \ddot{\xi}_3 \\ \ddot{\xi}_2 \\ \dot{\lambda}
\end{bmatrix}
=
$$

$$
\begin{bmatrix}
0 & 0 & I_{n-q} & 0 & 0 & 0 & 0 \\
-S_1 & -S_2 & 0 & 0 & 0 & 0 & 0 \\
-K_{11}+D_{12}S_1 & -K_{12}+D_{12}S_2 & -D_{11} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & I_{q_p} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I_{q_p} & 0 & 0 \\
S_1 & S_2 & 0 & 0 & 0 & I_{q_s} & 0 \\
-\overline{K}_{21}+\overline{D}_{22}S_1 & -\overline{K}_{22}+\overline{D}_{22}S_2 & -\overline{D}_{21} & 0 & 0 & 0 & I_q
\end{bmatrix}
\begin{bmatrix}
\xi_1 \\ \xi_2 \\ \dot{\xi}_1 \\ \xi_3 \\ \dot{\xi}_3 \\ \dot{\xi}_2 \\ \lambda
\end{bmatrix}
+
\begin{bmatrix}
0 \\ 0 \\ f_1 \\ 0 \\ 0 \\ 0 \\ \overline{f}_2
\end{bmatrix}
\qquad (2.4.23)
$$

Premultiplying both sides of (2.4.23) by the following nonsingular matrix

$$
\mathrm{P} =
\begin{bmatrix}
I_{n-q} & 0 & 0 & 0 \\
0 & I_{q_s} & 0 & 0 \\
M_{11}^{-1}M_{12}S_1 & M_{11}^{-1}M_{12}S_2 & M_{11}^{-1} & 0 \\
0 & 0 & 0 & I_{2q+q_p}
\end{bmatrix}
\qquad (2.4.24)
$$

results in

$$
\begin{bmatrix}
I_{n-q} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & I_{q_s} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_{q_p} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0_{q_p} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0_{q_p} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0_{q_s} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0_q
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_1 \\ \dot{\xi}_2 \\ \ddot{\xi}_1 \\ \dot{\xi}_3 \\ \ddot{\xi}_3 \\ \ddot{\xi}_2 \\ \dot{\lambda}
\end{bmatrix}
=
$$

$$\begin{bmatrix} 0 & 0 & I_{n-q} & 0 & 0 & 0 & 0 \\ -S_1 & -S_2 & 0 & 0 & 0 & 0 & 0 \\ A_1 & A_2 & A_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{q_p} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{q_p} & 0 & 0 \\ S_1 & S_2 & 0 & 0 & 0 & I_{q_v} & 0 \\ -\bar{K}_{21}+\bar{D}_{21}S_1 & -\bar{K}_{22}+\bar{D}_{22}S_2 & -\bar{D}_{21} & 0 & 0 & 0 & I_q \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \\ \xi_3 \\ \dot{\xi}_3 \\ \dot{\xi}_2 \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_{11}^{-1}f_1 \\ 0 \\ 0 \\ 0 \\ \bar{f}_2 \end{bmatrix} \qquad (2.4.25)$$

where

$$A_1 = M_{11}^{-1}(D_{12}S_1 - K_{11} - M_{12}\dot{S}_2 S_1) \qquad (2.4.26)$$

$$A_2 = M_{11}^{-1}(D_{12}S_2 - K_{12} - M_{12}\dot{S}_2 S_2) \qquad (2.4.27)$$

$$A_3 = M_{11}^{-1}(M_{12}S_1 - D_{11}) \qquad (2.4.28)$$

Then the first three rows of (2.4.25) offer the dynamic part of the normal form representation as expressed by the following:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \ddot{\xi}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & I_{n-q} \\ -S_1 & -S_2 & 0 \\ A_1 & A_2 & A_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_{11}^{-1} \end{bmatrix} f_1 \qquad (2.4.29)$$

Expanding the last two rows of (2.4.23) results in:

$$\begin{bmatrix} 0 & 0 & 0 \\ -\bar{M}_{22}S & -\bar{M}_{22}S_2 & \bar{M}_2 \end{bmatrix} \begin{bmatrix} \ddot{\xi}_1 \\ \ddot{\xi}_2 \\ \ddot{\xi}_1 \end{bmatrix} =$$

$$\begin{bmatrix} S_1 & S_2 & 0 \\ -\bar{K}_{21}+\bar{D}_{22}S_1 & -\bar{K}_{22}+\bar{D}_{22}S_2 & -\bar{D}_{21} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} \dot{\xi}_2 \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{f}_2 \end{bmatrix} \qquad (2.4.30)$$

Simplification of equation (2.4.30) yields the algebraic part of the normal form representation as the following

$$\begin{bmatrix} \dot{\xi}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -S_1 & -S_2 & 0 \\ E_1 & E_2 & E_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ M_{21}M_{11}^{-1} & -I \end{bmatrix} \begin{bmatrix} f_1 \\ \bar{f}_2 \end{bmatrix} \qquad (2.4.31)$$

where

$$E_1 = \overline{M}_{21} A_1 + \overline{M}_{22} S_2 S_1 + \overline{K}_{21} - \overline{D}_{22} S_1 \qquad (2.4.32)$$

$$E_2 = \overline{M}_{21} A_2 + \overline{M}_{22} S_2 S_2 + \overline{K}_{22} - \overline{D}_{22} S_2 \qquad (2.4.33)$$

$$E_3 = \overline{M}_{21} A_3 - \overline{M}_{22} S_1 + \overline{D}_{21} \qquad (2.4.34)$$

Moreover, the output equation from (2.3.33) of the last section

$$y = \begin{bmatrix} C_p & C_v & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix}$$

can be rewritten as

$$y = (C_l T)(T^T z) + (C_z T)(T^T \dot{z})$$

$$= (C_l T)\xi + (C_v T)\dot{\xi}$$

$$= \begin{bmatrix} C_1 & C_2 & C_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} + \begin{bmatrix} C_4 & C_5 & C_6 \end{bmatrix} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix}$$

Substituting $\xi_3 = 0$ from equation (2.4.13) into the above equation results in

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} C_4 & C_5 \end{bmatrix} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} \qquad (2.4.35)$$

Further substituting $\dot{\xi}_2 = \begin{bmatrix} -S_1 & -S_2 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix}$ from (2.4.11) into the above equation (2.4.35)

results in the output equation of the normal form representation as the following:

$$y = \begin{bmatrix} C_1 - C_5 S_1 & C_2 - C_5 S_2 & C_4 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} \qquad (2.4.36)$$

The above development generated a normal form for a system with both holonomic and nonholonomic constraints. This coordinate transformation procedure can be performed on systems with only holonomic constraints to derive normal form representations for this kind of system. It can also be used to derive normal forms for systems with only nonholonomic constraints. The results of the derivation of normal forms for the aforementioned three kinds of constrained linear mechanical systems can be summarized as follows:

Theorem 2.4.1 - The constrained mechanical systems of form (2.3.32)-(2.3.33) can always be transformed into one of the following forms:

(a) in the case of only holonomic constraints:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & I_{n-q} \\ -M_{11}^{-1}K_{11} & -M_{11}^{-1}D_{11} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ M_{11}^{-1} \end{bmatrix} f_1 \qquad (2.4.37)$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \qquad (2.4.38)$$

$$\xi_2 = \dot{\xi}_2 = 0 \qquad (2.4.39)$$

$$\lambda = (K_{21} - M_{21}M_{11}^{-1}K_{11})\xi_1 + (D_{21} - M_{21}M_{11}^{-1}K_{11})\dot{\xi}_1 + M_{21}M_{11}^{-1}f_1 - f_2 \qquad (2.4.40)$$

(b) in the case of only nonholonomic constraints:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & I_{n-q} \\ -S_1 & -S_2 & 0 \\ A_1 & A_2 & A_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_{11}^{-1} \end{bmatrix} f_1 \qquad (2.4.41)$$

$$y = \begin{bmatrix} C_1 - C_4S_1 & C_2 - C_4S_2 & C_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \qquad (2.4.42)$$

$$\begin{bmatrix} \dot{\xi}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -S_1 & -S_2 & 0 \\ E_1 & E_2 & E_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ M_{21}M_{11}^{-1} & -I \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \qquad (2.4.43)$$

(c) in the case of both holonomic and nonholonomic constraints.

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \ddot{\xi}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & I_{n-q} \\ -S_1 & -S_2 & 0 \\ A_1 & A_2 & A_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M_{11}^{-1} \end{bmatrix} f_1 \tag{2.4.44}$$

$$y = \begin{bmatrix} C_1 - C_3 S_1 & C_2 - C_3 S_2 & C_4 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} \tag{2.4.45}$$

$$\xi_3 = \dot{\xi}_3 = 0 \tag{2.4.46}$$

$$\begin{bmatrix} \dot{\xi}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} -S_1 & -S_2 & 0 \\ E_1 & E_2 & E_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ M_{21} M_{11}^{-1} & -I \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{2.4.47}$$

where ξ_i's are defined in equation (2.4.1) and other matrices in the above normal forms are defined as follows:

1) for case (a) and (b).

$$T^T \Lambda T = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}, \qquad \Lambda \text{ denotes M, D, or K}$$

$$T^T f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \; C_p T = \begin{bmatrix} C_1 & C_2 \end{bmatrix}, C_v T = \begin{bmatrix} C_3 & C_4 \end{bmatrix}$$

2) for case (c):

$$T^T \Lambda T = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \Lambda_{13} \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} \\ \Lambda_{31} & \Lambda_{32} & \Lambda_{33} \end{bmatrix} \; \Lambda := M, D, K, \; T^T f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$\tilde{f}_2 = \begin{bmatrix} f_2 \\ f_3 \end{bmatrix}, \quad C_p T = \begin{bmatrix} C_1 & C_2 & C_3 \end{bmatrix}, \; C_v T = \begin{bmatrix} C_4 & C_5 & C_6 \end{bmatrix}$$

3) for cases (b) and (c)

$$A_1 = M_{11}^{-1}(D_{12} S_1 - K_{11} - M_{12} S_2 S_1)$$

$$A_2 = M_{11}^{-1}(D_{12} S_2 - K_{12} - M_{12} S_2 S_2)$$

$$A_3 = M_{11}^{-1}(M_{12} S_1 - D_{11})$$

4) for case (b).

$$E_1 = M_{21} A_1 + M_{22} S_2 S_1 - K_{21} - D_{22} S_1$$

$$E_2 = M_{21} A_2 + M_{22} S_2 S_2 + K_{22} - D_{22} S_2$$

$$E_3 = M_{21} A_3 - M_{22} S_1 + D_{21}$$

5) for case (c):

$$E_1 = \overline{M}_{21} A_1 + \overline{M}_{22} S_2 S_1 + \overline{K}_{21} - \overline{D}_{22} S_1$$

$$E_2 = \overline{M}_{21} A_2 + \overline{M}_{22} S_2 S_2 + \overline{K}_{22} - \overline{D}_{22} S_2$$

$$E_3 = \overline{M}_{21} A_3 - \overline{M}_{22} S_1 + \overline{D}_{21}$$

2.5 Regularity of Constrained Linear Mechanical Systems

As defined in definition 2.4 of section 2.1, regularity refers to the existence and uniqueness of a solution. Obviously regularity is an important property of the systems being studied in this thesis. Working with the following special form representation of linearized constrained mechanical systems

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \overline{S} & \overline{N} & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} \tag{2.5.1}$$

$$y = \begin{bmatrix} C_p & C_v & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} \tag{2.5.2}$$

we can come up with a theorem as follows:

Theorem 2.5.1 Constrained linear mechanical systems as described by equations (2.5.1) and (2.5.2) are always regular, i.e.

$$\begin{vmatrix} sI_n & -I_n & 0 \\ K & sM + D & -J^T \\ -\overline{S} & -\overline{N} & 0 \end{vmatrix} \not\equiv 0, s \in C \tag{2.5.3}$$

Proof: A proof using the shuffle algorithm (Luenberger,1978) is provided in Appendix A of this thesis

2.6 Controllability of Constrained Linear Mechanical Sysfems

As defined in section 2.1, the necessary and sufficient condition for reachable controllability is one of the two conditions for complete controllability. Consequently, complete controllability is definitely a stronger condition than reachable controllability. A system that has reachable controllability does not necessarily have complete controllability. A system that have complete controllability must have reachable controllability. It can be shown that the stronger controllability property is not possessed by constrained linear mechanical systems. Being a special class of linear descriptor systems, constrained linear mechanical systems or linear mechanical descriptor systems can have reachable controllability at best.

Theorem 2.6.1 Linear mechanical descriptor systems as described by equations (2.5.1) and (2.5.2) are always not completely controllable or do not have C-controllability.

Proof: The proof is easily obtained by showing that the second rank condition in the definition of complete controllability is not satisfied, i.e., [E B] is not of full row rank

First we need to rewrite (2.5.1) to show what B and E are

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ \bar{S} & \bar{N} & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} f$$

$$E = \begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}$$

Now it is obvious that

$$\text{rank [E B]} = \text{rank} \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & M & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} \neq \text{number of rows in E or B}$$

The second rank condition is not satisfied and this completes the proof

2.7 Observability of and Observer Existence Conditions for Constrained Linear Mechanical Systems

Any observer-based fault detection and identification schemes inevitably relies on system observability conditions of one kind or another. I now present the following theorem regarding the infinite observability of linearized constrained mechanical systems:

Theorem 2.7.1 Linear mechanical descriptor systems as described by equations (2.5.1) and (2.5.2) are always infinitely unobservable or do not possess impulse observability.

Proof: see Appendix B of this thesis.

Shin and Kabamba(1988) noticed that when constrained forces are not directly or indirectly measurable, a constrained mechanical system is not infinitely observable. The mathematical proof in Appendix B confirms this physical explanation

Infinite observability is a desirable feature as far as unknown-input decoupled observer design is concerned. Some general results of conventional observers for descriptor systems with unknown inputs are provided by Hou and Muller(1992a). The existence conditions of the unknown-input observer has the nature of the infinite observability and therefore can not be met in constrained mechanical systems considered here. The design of unknown-input observer must be based on weaker or alternative observability conditions. Designing unknown-input observers is an essential part of any observer-based FDI schemes. An unknown input observer (UIO) design method is presented in the next chapter.

Incidentally, the observer existence condition for constrained linear mechanical systems driven by totally known inputs is very simple. Hou and et. al (Hou and et al.,1993) pointed out that finite observability of the system is a necessary and sufficient condition. They also proved that finite observability of the descriptor system is equivalent to the observability of the corresponding conventional system in the minimal coordinates. Since this thesis is most concerned with unknown input observer based FDI, their condition is not really useful. Necessary and sufficient conditions for the existence of unknown-input observers remain to be found in a subsequent chapter.

2.8 Summary

In this chapter, we enriched and extended the discussion and analysis of constrained dynamic systems and constrained mechanical systems by some previous researchers such as Hou, et al. (1993). We recognized and classified constrained mechanical systems as a special kind of singular or descriptor systems. Starting with the standard nonlinear Lagrange equations model of constrained mechanical systems, we first obtained a linearized model of the system by using the standard Taylor series expansion technique. We were able to rewrite the linear model in a generalized state-space format. Then we performed a nonsingular transformation on the constraint equations and obtained a special descriptor form representation. This transformation process generated a nonsingular (orthogonal) matrix which we subsequently used in performing a coordinate transformation and deriving normal forms for mechanical systems with holonomic and/or nonholonomic constraints. We used a numerical example in demonstrating how one can perform the important transformation. The resultant dynamic subsystem in the normal form of a constrained mechanical system leads our subsequent studies in the following chapter to the domain of linear system theory. In the last few sections of this chapter, we identified and discussed some rather special properties of constrained mechanical systems such as their lack of infinite observability and complete controllability. We pointed out that lack of infinite observability restricted our choices of approaches to observer design and fault detection and identification. By doing so we built a bridge between this chapter and the next one, which presents the design of an observer that is capable of estimating the state of a system driven by both known and unknown inputs.

Chapter 3

An Unknown Input Observer Based Fault Detection and Identification Method

3.1 Introduction

As discussed in chapter one, the need for safe and reliable operation of complex engineering processes at reasonable cost has been promoting research and investigation into the problem of fault detection, identification, and accommodation (FDIA). Among the various FDIA techniques, there is a class of model-based approaches that are commonly referred to as the analytical redundancy techniques. Detailed survey of these methods could be found in Willsky (1976), Isermann (1984), Merrill (1985), and Frank (1990). Since the introduction of dedicated observer scheme (DOS) by Clark (1978), more sophisticated approaches based on it utilizing some detection function or statistical tests have been proposed.

One of the major difficulties in the application of model-based techniques to practical FDIA cases is the problem of plant uncertainties or parameter variations. In such situations there usually exists a need for a robust FDIA methodology. A number of different approaches to robust FDIA problems have been proposed. One such approach is a sensitivity discriminating observer scheme proposed by Frank and Keller (1980). Another approach dealing with uncertainties is the threshold selector method proposed by Emami-Naeini, et al. (1988).

Recently, there have been some studies in the area of FDIA based on the theory of unknown input observers (UIO). A survey of the UIO-based approaches can be found in Frank (1990). Several somewhat different UIO design methods have been proposed by Kudva (1980), Kurek (1983), Wang et al. (1975), Yang and Wilde (1988), Guan and Saif (1991), and Hou and Muller (1992). The UIO theory has been utilized for actuator fault detection and isolation by Viswanadham and Srichander (1987), and Park and Stein (1988).

UIO design has been an active area of research in the past several years due to its widespread applications. UIO's are primarily designed to accommodate unknown exogenous disturbances in the dynamics of the plant. Conventional observers that reconstructs the state vector under the assumption that all inputs are known have been used in state feedback control of various systems. This traditional approach of control neglected the presence of certain uncertainties (such as inaccessible inputs and plant disturbances) and often is not sufficiently useful for fault detection and identification purpose. Because most uncertainties and plant faults can be modeled as unknown inputs to the system, designing unknown input observers (UIO) is of tremendous use for robust control, fault detection, identification, and accommodation (FDIA).

Basically, there are two types of UIO design methods. The first category of approaches includes a number of attempts that assumed some *a priori* information about the unmeasurable inputs to the system. Specifically, Johnson (1975) assumes a polynomial approximation to these inputs, and in Meditch & Hostetter (1974), it is assumed that the unknown inputs can be modeled as the response of a known dynamic system represented by a constant coefficient differential equation. The other category of UIO studies assumes no knowledge of the inaccessible inputs. Among the more recent works are those of Yang & Wilde (1988), Guan and Saif (1991), and Hou & Muller (1992). Yang & Wilde proposes a full-order observer that is claimed to have somewhat better rate of convergence than a reduced-order observer. Although they claim that they use straightforward matrix calculations, their procedure involves singular value decomposition or Jordan form transformation. Their method also requires solving a system of linear equations that has more unknowns than equations. In the work of Hou and Muller (1992), a reduced-order observer and a minimal-order observer are derived via a technique of coordinate transformation. The derivation is rather mathematically involved and hard to understand. In this chapter, we propose a mathematically simple and computationally efficient unknown input observer

(UIO) design method. This method is inspired by and owes its merits to the early work of Guan and Şaif (1991). In the following few sections of this chapter, the UIO design is discussed and some modification of the approach of Guan and Saif (1991) is made to make the UIO design more systematic. The first step of the procedure is formulating the problem as a linear time-invariant system with unknown inputs. The second step is specifying the assumptions used in UIO design. The third step involves performing a nonsingular transformation on the partitioned system and actually deriving an observer for one of the three reduced order subsystems. It turns out that states of the other two subsystems have direct algebraic relationships with the output of the system which is assumed to be available (measurable) for observer design purpose. Thus the combined state of the whole system can be estimated using a single conventional observer. Furthermore, a necessary and sufficient condition for the existence of an UIO is presented and proved in this chapter. This condition can be expressed in terms of the matrices in the linear time-invariant representation of the system for the convenience of checking if the condition is met. Finally, methods for detecting and identifying actuator faults and sensor faults are presented in the last part of this chapter. The methods are based on modeling actuator faults as unknown inputs to the dynamic equations of the system and modeling sensor faults as unknown inputs to an augmented system of which sensor faults are part of the state. Both actuator fault detection method and sensor fault detection method rely on state estimation which is accomplished via the unknown input observer.

3.2 System Representation and its Observable Canonical Form

System description and modeling has been discussed in the previous chapter and will be discussed further in the last part of this chapter. In the next few sections we are only concerned with deriving an UIO. For this purpose we assume that all the necessary linearization and transformations have been performed and our system representation has resulted in the simplest form of all dynamic system representations, namely linear time-invariant systems. These systems can be assumed to be driven by partially unknown inputs which may be used to represent plant faults and parameter uncertainties. The state-space formulation can be given as follows:

$$\dot{\bar{x}} = \bar{A}\,\bar{x} + \bar{B}\,u + \bar{D}\,d \qquad (3.2.1)$$

$$y = \bar{C}\,\bar{x} \qquad (3.2.2)$$

Without loss of generality, the concerned system can be written in the following observable canonical form:

$$\dot{x} = Ax + Bu + Dd \qquad (3.2.3)$$

$$y = Cx = \begin{bmatrix}0 & I\end{bmatrix}x \qquad (3.2.4)$$

where

$A \in \Re^{n \times n}$, $B \in \Re^{n \times q}$, $C \in \Re^{p \times n}$, $D \in \Re^{n \times m}$

$x \in \Re^{n \times 1}$, n: number of state variables

$u \in \Re^{q \times 1}$, q: number of known inputs

$d \in \Re^{m \times 1}$, m: number of unknown inputs

$y \in \Re^{p \times 1}$, p: number of outputs

I is an identity matrix of order $p \times p$

Remark: If C is of full row rank, there always exists a similarity transformation that can bring the representation in (3.2.1) & (3.2.2) into its observable canonical form in (3.2.3) & (3.2.4). Details of this procedure and the proof of this claim can be found in the book of Chen(1984)

3.3 Assumptions for the Design of UIO's

Three assumptions are made in the rest of this work. These assumptions have been used implicitly or explicitly in all the works on UIO theory and design. As can be explained later, they are not restrictive assumptions:

Assumption 1

The measurement matrix $\overline{C}$ in (3.2.2) is assumed to be of full row rank, i.e.

$$\text{rank } \overline{C} = p \tag{3.3.1}$$

If the measurement matrix $\overline{C}$ is not of full rank, then there exists at least one redundant output. This redundancy can be eliminated by redefining the output vector y and the measurement matrix $\overline{C}$ such that the new outputs are linearly independent. Therefore, this is not a restrictive assumption.

Assumption 2

The D matrix in (3.2.3) is assumed to be of full column rank, i.e.

$$\text{rank } D = m \tag{3.3.2}$$

If D is not of full rank, it can always be decomposed as a product of two full rank matrices via the following proposition:

Proposition 3.3.1 Any $p \times q$ matrix A, whose rank is r can be decomposed as follows:

$$A = BC$$

where

B is a $p \times r$ full rank matrix

C is an $r \times q$ full rank matrix

Proof See the proof of proposition 2 in Saif and Guan (1993)

Thus, D can be decomposed as

$$D=\bar{D}\bar{N} \tag{3.3.3}$$

where

$\bar{D}$ has full column rank

$\bar{N}$ has full row rank

and a full rank new D and a new d for (3.2.3) can be defined as

$$D^{*} = \bar{D}$$

$$d^{*} = \bar{N}\ d$$

In the early work of Kudva et al. (1980), a necessary condition for the existence of any unknown input observers for the system described by (3.2.3) and (3.2.4) is proposed and is subsequently used explicitly or implicitly by many others. It can be stated as follows:

Assumption 3

A necessary condition for the existence of a stable unknown input observer for the linear dynamic system described by (3.2.3) and (3.2.4) is that the number of linearly independent outputs is greater than or equal to the number of unknown inputs. i.e.,

$$\text{rank}(CD) = \text{rank}(D) = m\ , \quad \text{with}\ \ m \le p \tag{3.3.4}$$

Proof: See theorem 1 in Saif and Guan(1993)

3.4 Unknown Input Observer(UIO) Design

First, we apply the partition technique developed by Saif and Guan (1993) to divide the dynamic system in (3.2.3)-(3.2.4) into three subsystems:

$$\begin{bmatrix} \dot{x}_{1_{(n-p)\times 1}} \\ \dot{x}_{2_{(p-m)\times 1}} \\ \dot{x}_{3_{m\times 1}} \end{bmatrix} = \begin{bmatrix} A_{1_{(n-p)\times n}} \\ A_{2_{(p-m)\times n}} \\ A_{3_{m\times n}} \end{bmatrix} \begin{bmatrix} x_{1_{(n-p)\times 1}} \\ x_{2_{(p-m)\times 1}} \\ x_{3_{m\times 1}} \end{bmatrix} + \begin{bmatrix} B_{1_{(n-p)\times q}} \\ B_{2_{(p-m)\times q}} \\ B_{3_{m\times q}} \end{bmatrix} \mathrm{u} + \begin{bmatrix} D_{1_{(n-p)\times m}} \\ D_{2_{(p-m)\times m}} \\ D_{3_{m\times m}} \end{bmatrix} \mathrm{d} \qquad (3.4.1)$$

$$\mathrm{y} = \begin{bmatrix} 0_{p\times(n-p)} & I_{p\times p} \end{bmatrix} \begin{bmatrix} x_{1_{(n-p)\times 1}} \\ x_{2_{(p-m)\times 1}} \\ x_{3_{m\times 1}} \end{bmatrix} \qquad (3.4.2)$$

As shown in the later part of this section, the UIO design procedure involves using a nonsingular transformation matrix which contains the inverse of D_3. In general, a simple straightforward partition of the observable canonical form represented by (3.2.3) and (3.2.4) does not necessarily result in an invertible D_3. A procedure is needed to deal with the lack of invertibility of D_3. It turns out that this can easily be accomplished by reordering state variables in the observable canonical form representation.

Since $\begin{bmatrix} D_2 \\ D_3 \end{bmatrix}_{p\times m}$ is of full column rank by assumption 3 in the last section, it must have a m × m submatrix whose determinant is nonzero. This submatrix is therefore invertible and can be defined as the new D_3. The new D_3 just contains different rows of $\begin{bmatrix} D_2 \\ D_3 \end{bmatrix}_{p\times m}$ than the old D_3 does. It can be obtained by switching the rows of $\begin{bmatrix} D_2 \\ D_3 \end{bmatrix}_{p\times m}$. Switching the rows of $\begin{bmatrix} D_2 \\ D_3 \end{bmatrix}_{p\times m}$ is equivalent to reordering the last p state variables in (3.4.1). The first (n-p) state variables are contained in D_1 and therefore does not need reordering. Thus a nonsingular or invertible D_3 can always be obtained by reordering state variables. What must be pointed out is that reordering state variables also affects the output equation and thus the measurement matrix C. As a matter of fact, reordering state variables result in column exchanges in C. Since only the last p state variables are reordered, column switchings in $C_{p\times n}$

are limited to the last p columns. Because the last p columns of $C_{p \times n}$ in the canonical form (3.2.4) is an identity submatrix I_p which is absolutely nonsingular, the new submatrix resulting from switching columns in I_p has to remain nonsingular. This invertible submatrix can be denoted as C_p. Then D_3 can be assumed nonsingular while C is of the form $\begin{bmatrix} 0 & C_p \end{bmatrix}$.

It is worth noting that reordering state variables is equivalent to performing a similarity transformation on the observable form representation. The observable form representation itself can also be obtained by performing a similarity transformation on the original state-space model.

Therefore, without loss of generality, the system may be assumed to be of the following form:

$$\dot{x} = Ax + Bu + Dd = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} x + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} u + \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} d \qquad (3.4.3)$$

$$y = Cx = \begin{bmatrix} 0 & C_p \end{bmatrix} x \qquad (3.4.4)$$

where

$A \in \Re^{n \times n}$, $B \in \Re^{n \times q}$, $C \in \Re^{p \times n}$, $D \in \Re^{n \times m}$

$x \in \Re^{n \times 1}$, n: number of state variables

$u \in \Re^{q \times 1}$, q: number of known inputs

$d \in \Re^{m \times 1}$, m: number of unknown inputs

$y \in \Re^{p \times 1}$, p: number of outputs

C_p is an invertible matrix of order $p \times p$

The output equation (3.4.2) can be partitioned as:

$$\mathrm{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & C_p \end{bmatrix} \mathrm{x} = \mathrm{C}_\mathrm{p} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \tag{3.4.5}$$

The inverse of C_p can be defined as:

$$\mathrm{C}_\mathrm{p}^{-1} = \begin{bmatrix} C_{p1} \\ C_{p2} \end{bmatrix} \tag{3.4.6}$$

Then equation (3.4.3) can be rewritten as:

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} C_{p1} \\ C_{p2} \end{bmatrix} \mathrm{y} \tag{3.4.7}$$

or

$$\mathrm{x}_2 = \mathrm{C}_{\mathrm{p}1}\, \mathrm{y} \tag{3.4.8}$$

$$\mathrm{x}_3 = \mathrm{C}_{\mathrm{p}2}\, \mathrm{y} \tag{3.4.9}$$

The following matrix operator can be defined:

$$\mathrm{T} = \begin{bmatrix} I & 0 & -D_1 D_3^{-1} \\ 0 & I & -D_2 D_3^{-1} \\ 0 & 0 & I \end{bmatrix} \tag{3.4.10}$$

Post-multiplying both sides of equation (3.4.1) with the above operator results in:

$$\begin{bmatrix} \dot{x}_1 - D_1 D_3^{-1} \dot{x}_3 \\ \dot{x}_2 - D_2 D_3^{-1} \dot{x}_3 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} A_1 - D_1 D_3^{-1} A_3 \\ A_2 - D_2 D_3^{-1} A_3 \\ A_3 \end{bmatrix} \mathrm{x} + \begin{bmatrix} B_1 - D_1 D_3^{-1} B_3 \\ B_2 - D_2 D_3^{-1} B_3 \\ B_3 \end{bmatrix} \mathrm{u} + \begin{bmatrix} 0 \\ 0 \\ D_3 \end{bmatrix} \mathrm{d} \tag{3.4.11}$$

Substituting (3.4.8) and (3.4.9) into the first two "rows" of (3.4.11) yields:

$$\dot{x}_1 - D_1 D_3^{-1} C_{p2}\,\dot{y} = \bar{A}_1 x + \bar{B}_1 u \tag{3.4.12}$$

and

$$C_{p1}\,\dot{y} - D_2 D_3^{-1} C_{p2}\,\dot{y} = \bar{A}_2 x + \bar{B}_2 u \tag{3.4.13}$$

where

$$\bar{A}_i = A_i - D_i D_3^{-1} A_3$$
$$\bar{B}_i = B_i - D_i D_3^{-1} B_3$$

Partitioning $\bar{A}_i$ as

$$\bar{A}_i = \begin{bmatrix} \bar{A}_{i1} & \bar{A}_{i2} & \bar{A}_{i3} \end{bmatrix} \qquad i = 1, 2, 3 \tag{3.4.14}$$

where

$$\bar{A}_{ij} = A_{ij} - D_i D_3^{-1} A_{3j} \qquad j = 1, 2, 3 \tag{3.4.15}$$

and A_{ij} 's are elements of partioned A matrix of the following form

$$A = \begin{bmatrix} A_{11_{(n-p)\times(n-p)}} & A_{12_{(n-p)\times(p-m)}} & A_{13_{(n-p)\times m}} \\ A_{21_{(p-m)\times(n-p)}} & A_{22_{(p-m)\times(p-m)}} & A_{23_{(p-m)\times m}} \\ A_{31_{m\times(n-p)}} & A_{32_{m\times(p-m)}} & A_{33_{m\times m}} \end{bmatrix}$$

Now using the partioned matrices $\bar{A}_{ij}$ in (3.4.12) and (3.4.13) will result in:

$$\dot{x}_1 = \bar{A}_{11} x_1 + r \tag{3.4.16}$$

and

$$z = \bar{A}_{21} x_1 \tag{3.4.17}$$

where

$$r = (\bar{A}_{12} C_{p1} + \bar{A}_{13} C_{p2}) y + D_1 D_3^{-1} C_{p2}\,\dot{y} + \bar{B}_1 u \tag{3.4.18}$$

and

$$z = (C_{p1} - D_2 D_3^{-1} C_{p2})\,\dot{y} - (\bar{A}_{22} C_{p1} + \bar{A}_{23} C_{p2}) y - \bar{B}_2 U \tag{3.4.19}$$

According to observer theory(Chen, 1984), the state of the dynamic system represented by (3.4.16) and (3.4.17) can be estimated by a Luenberger observer. The dynamics of this reduced-order observer is given by:

$$\dot{\hat{x}}_1 = \bar{A}_{11}\hat{x}_1 + r + M(z - \bar{A}_{21}\hat{x}_1) \tag{3.4.20}$$

where M is the observer's gain. Substituting for r and z into (3.4.20):

$$\begin{aligned}\dot{\hat{x}}_1 &= \bar{A}_{11}\hat{x}_1 + (\bar{A}_{12}C_{p1} + \bar{A}_{13}C_{p2})y + D_1D_3^{-1}C_{p2}\dot{y} + \bar{B}_1u + M\{(C_{p1} - D_2D_3^{-1}C_{p2})\dot{y} \\ &\quad - (\bar{A}_{22}C_{p1} + \bar{A}_{23}C_{p2})y - \bar{B}_2u - \bar{A}_{21}\hat{x}_1\} \\ &= (\bar{A}_{11} - M\bar{A}_{21})\hat{x}_1 + \{(\bar{A}_{12}C_{p1} + \bar{A}_{13}C_{p2}) - M(\bar{A}_{22}C_{p1} + \bar{A}_{23}C_{p2})\}y \\ &\quad + (\bar{B}_1 - M\bar{B}_2)u + \{D_1D_3^{-1}C_{p2} - M(C_{p1} - D_2D_3^{-1}C_{p2})\}\dot{y}\end{aligned} \tag{3.4.21}$$

Equation (3.4.21) contains the derivative of the output which is not available for direct measurement. This problem can be dealt with by defining a new variable w as follows:

$$w = \hat{x}_1 - \{D_1D_3^{-1}C_{p2} - M(C_{p1} - D_2D_3^{-1}C_{p2})\}y \tag{3.4.22}$$

Rewriting (3.4.21) in terms of the new variable w will result in:

$$\dot{w} = Fw + Ey + Lu \tag{3.4.23}$$

where

$$F = (\bar{A}_{11} - M\bar{A}_{21}) \tag{3.4.24}$$

$$\begin{aligned}E &= (\bar{A}_{11} - M\bar{A}_{21})\{D_1D_3^{-1}C_{p2} - M(C_{p1} - D_2D_3^{-1}C_{p2})\} \\ &\quad + \{(\bar{A}_{12}C_{p1} + \bar{A}_{13}C_{p2}) - M(\bar{A}_{22}C_{p1} + \bar{A}_{23}C_{p2})\}\end{aligned} \tag{3.4.25}$$

$$L = (\bar{B}_1 - M\bar{B}_2) \tag{3.4.26}$$

The following theorem will conclude and summarize UIO design.

Theorem 3.4.1 If the pair $\{\bar{A}_{11}, \bar{A}_{21}\}$ is observable, the state of the dynamic system described by (3.4.1) and (3.4.2) can be estimated by using the UIO proposed in (3.4.23)-(3.4.26). The estimate of the state is given by:

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} w + \begin{bmatrix} N \\ C_{p1} \\ C_{p2} \end{bmatrix} y \tag{3.4.27}$$

where

$$N = D_1 D_3^{-1} C_{p2} - M(C_{p1} - D_2 D_3^{-1} C_{p2}) \tag{3.4.28}$$

In addition, all the eigenvalues of F can be placed at any desired location. The proof of the above theorem has been implicitly given in the foregoing discussion.

3.5 Necessary and Sufficient Conditions for the Existence of UIO's

Given the linear time-invariant dynamic system with partially unknown inputs as described in the previous sections:

$$\dot{x} = Ax + Bu + Dd \tag{3.5.1}$$

$$y = Cx \tag{3.5.2}$$

where

$A \in \Re^{n \times n}$, $B \in \Re^{n \times q}$, $C \in \Re^{p \times n}$, $D \in \Re^{n \times m}$

$x \in \Re^{n}$, n: number of state variables

$u \in \Re^{q}$, q: number of known inputs

$d \in \Re^{m}$, m: number of unknown inputs

$y \in \Re^{p}$, p: number of outputs

there exists a necessary and sufficient condition for the existence of unknown input observers. Since different researchers in this area use different methods to design different UIO's, there exist quite a few seemingly different necessary and sufficient conditions. After close examination of these competing conditions it is found that these conditions are virtually equivalent to each other. The generally accepted format of the condition can be stated in the following theorem:

Theorem 3.5.1 --- A Necessary and Sufficient Condition for the Existence of UIO's

A necessary and sufficient condition for the existence of an UIO for the system described by (3.5.1) and (3.5.2) is that

$$\text{rank}\begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix} = n + m \qquad \forall\ s \in C \tag{3.5.3}$$

It has to be pointed out that the concerned system such as (3.5.1)-(3.5.2) has been assumed to satisfy the minimum necessary conditions expressed in Assumption 1, Assumption 2, and Assumption 3 of section 3.3.

Proof: The above theorem can be proved indirectly by showing its equivalence to Theorems 3.4.1 which states that the necessary and sufficient condition as the observability of the pair $\{\bar{A}_{11}, \bar{A}_{21}\}$, i.e.,

$$\text{rank}\begin{bmatrix} sI_{n-p} - \bar{A}_{11} \\ \bar{A}_{21} \end{bmatrix} = n - p \qquad \forall\ s \in C \tag{3.5.4}$$

Define the following nonsingular matrix,

$$P = \begin{bmatrix} U & V \\ 0 & I_p \end{bmatrix} \tag{3.5.5}$$

where

$$\mathrm{U} = \begin{bmatrix} I_{n-p} & 0 & -D_1 D_3^{-1} \\ 0 & I_{p-m} & -D_2 D_3^{-1} \\ 0 & 0 & D_3^{-1} \end{bmatrix} \tag{3.5.6}$$

and

$$\mathrm{V} = \begin{bmatrix} \bar{A}_{12} & sD_1 D_3^{-1} + \bar{A}_{13} \\ -sI_{p-m} + \bar{A}_{22} & sD_2 D_3^{-1} + \bar{A}_{23} \\ D_3^{-1}\bar{A}_{32} & D_3^{-1}\bar{A}_{33} - sD_3^{-1} \end{bmatrix} \tag{3.5.7}$$

Note that C can be assumed to be of the form $\begin{bmatrix} 0 & I_p \end{bmatrix}$ without loss of generality. The matrices A, C and D can be partitioned as the following as they were partioned in section 3.4:

$$\mathrm{A} = \begin{bmatrix} A_{11_{(n-p)\times(n-p)}} & A_{12_{(n-p)\times(p-m)}} & A_{13_{(n-p)\times m}} \\ A_{21_{(p-m)\times(n-p)}} & A_{22_{(p-m)\times(p-m)}} & A_{23_{(p-m)\times m}} \\ A_{31_{m\times(n-p)}} & A_{32_{m\times(p-m)}} & A_{33_{m\times m}} \end{bmatrix} \tag{3.5.8}$$

$$\mathrm{D} = \begin{bmatrix} D_{1_{(n-p)\times m}} \\ D_{2_{(p-m)\times m}} \\ D_{3_{m\times m}} \end{bmatrix} \tag{3.5.9}$$

$$\mathrm{C} = \begin{bmatrix} 0_{(p-m)\times(n-p)} & I_{p-m} & 0_{(p-m)\times m} \\ 0_{m\times(n-p)} & 0_{m\times(p-m)} & I_m \end{bmatrix} \tag{3.5.10}$$

It is a known fact and a theorem in matrix theory that pre-multiplying a matrix with a nonsingular matrix preserves its rank. Now we can perform the following multiplication:

$$\operatorname{rank} \begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix} = \operatorname{rank} \mathrm{P} \begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix}$$

$$= \operatorname{rank} \begin{bmatrix} U & V \\ 0 & I_p \end{bmatrix} \begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix}$$

$$= \text{rank}\begin{bmatrix} U(sI_n - A) + VC & UD \\ C & 0 \end{bmatrix} \tag{3.5.12}$$

Now let us evaluate the upper two submatrices by defining

$$\text{W} = U(sI_n - A) \tag{3.5.13}$$

$$\text{X} = \text{VC} \tag{3.5.14}$$

$$\text{Y} = \text{W} + \text{X} \tag{3.5.15}$$

$$\text{Z} = \text{UD} \tag{3.5.16}$$

Then

$$\text{W}=\begin{bmatrix} I_{n-p} & 0 & -D_1D_3^{-1} \\ 0 & I_{p-m} & -D_2D_3^{-1} \\ 0 & 0 & D_{3_{m\times m}}^{-1} \end{bmatrix}\begin{bmatrix} sI_{n-p} - A_{11_{(n-p)\times(n-p)}} & -A_{12_{(n-p)\times(p-m)}} & -A_{13_{(n-p)\times m}} \\ -A_{21_{(p-m)\times(n-p)}} & sI_{p-m} - A_{22_{(p-m)\times(p-m)}} & -A_{23_{(p-m)\times m}} \\ -A_{31_{m\times(n-p)}} & -A_{32_{m\times(p-m)}} & sI_m - A_{33_{m\times m}} \end{bmatrix} =$$

$$\begin{bmatrix} sI_{n-p} - A_{11} + D_1D_3^{-1}A_{31} & -A_{12} + D_1D_3^{-1}A_{32} & -A_{13} - sD_1D_3^{-1} + D_1D_3^{-1}A_{33} \\ -A_{21} + D_2D_3^{-1}A_{31} & sI_{p-m} - A_{22} + D_2D_3^{-1}A_{32} & -A_{23} - sD_2D_3^{-1} + D_2D_3^{-1}A_{33} \\ -D_3^{-1}A_{31} & -D_3^{-1}A_{32} & sD_3^{-1} - D_3^{-1}A_{33} \end{bmatrix} \tag{3.5.17}$$

Recall that in equation (3.4.13) of section 3.4 we have defined

$$\bar{A}_{ij} = A_{ij} - D_iD_3^{-1}A_{3j} \qquad \text{i} = 1, 2, 3 \qquad \text{j} = 1, 2, 3 \tag{3.5.18}$$

Thus equation (3.5.17) can simplified as

$$\text{W}=\begin{bmatrix} sI_{n-p} - A_{11} + D_1D_3^{-1}A_{31} & -\bar{A}_{12} & -\bar{A}_{13} - sD_1D_3^{-1} \\ -A_{21} + D_2D_3^{-1}A_{31} & sI_{p-m} - \bar{A}_{22} & -\bar{A}_{23} - sD_2D_3^{-1} \\ -D_3^{-1}A_{31} & -D_3^{-1}A_{32} & sD_3^{-1} - D_3^{-1}A_{33} \end{bmatrix}$$

Now

$$
\mathrm{X} = \mathrm{VC} = \begin{bmatrix} \bar{A}_{12} & sD_1D_3^{-1} + \bar{A}_{13} \\ -sI_{p-m} + \bar{A}_{22} & sD_2D_3^{-1} + \bar{A}_{23} \\ D_3^{-1}\bar{A}_{32} & D_3^{-1}\bar{A}_{33} - sD_3^{-1} \end{bmatrix} \begin{bmatrix} 0_{(p-m)\times(n-p)} & I_{p-m} & 0_{(p-m)\times m} \\ 0_{m\times(n-p)} & 0_{m\times(p-m)} & I_m \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & \bar{A}_{12} & sD_1D_3^{-1} + \bar{A}_{13} \\ 0 & -sI_{p-m} + \bar{A}_{22} & sD_2D_3^{-1} + \bar{A}_{23} \\ 0 & D_3^{-1}A_{32} & D_3^{-1}A_{33} - sD_3^{-1} \end{bmatrix} \tag{3.5.19}
$$

Then we have

$$
\mathrm{Y} = \mathrm{W} + \mathrm{X} = \begin{bmatrix} sI_{n-p} - \bar{A}_{11} & 0 & 0 \\ -\bar{A}_{21} & 0 & 0 \\ -D_3^{-1}A_{31} & 0 & 0 \end{bmatrix} \tag{3.5.20}
$$

Furthermore

$$
\mathrm{Z} = \mathrm{UD} = \begin{bmatrix} I_{n-p} & 0 & -D_1D_3^{-1} \\ 0 & I_{p-m} & -D_2D_3^{-1} \\ 0 & 0 & D_3^{-1} \end{bmatrix} \begin{bmatrix} D_{1(n-p)\times m} \\ D_{2(p-m)\times m} \\ D_{3m\times m} \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_m \end{bmatrix} \tag{3.5.21}
$$

Therefore equation (3.5.12) can be rewritten as

$$
\operatorname{rank}\begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} Y & Z \\ C & 0 \end{bmatrix}
$$

$$
= \operatorname{rank}\begin{bmatrix} sI_{n-p} - \bar{A}_{11} & 0 & 0 & 0 \\ -\bar{A}_{21} & 0 & 0 & 0 \\ -D_3^{-1}A_{31} & 0 & 0 & I_m \\ 0 & I_{p-m} & 0 & 0 \\ 0 & 0 & I_m & 0 \end{bmatrix}
$$

$$= \text{rank}\begin{bmatrix} sI_{n-p} - \bar{A}_{11} & 0 & 0 & 0 \\ -\bar{A}_{21} & 0 & 0 & 0 \\ 0 & I_m & 0 & 0 \\ 0 & 0 & I_{p-m} & 0 \\ 0 & 0 & 0 & I_m \end{bmatrix}$$

$$= \text{rank}\begin{bmatrix} sI_{n-p} - \bar{A}_{11} \\ \bar{A}_{21} \end{bmatrix} + m + (p-m) + m \qquad (3.5\ 22)$$

Thus

$$\text{rank}\begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix} = n + m \quad \Leftrightarrow \quad \text{rank}\begin{bmatrix} sI_{n-p} - \bar{A}_{11} \\ \bar{A}_{21} \end{bmatrix} = n - p$$

This shows the equivalency between the two conditions and concludes the proof. This also concludes our discussion of unknown input observers.

3.6 Problem Formulation for Fault Detection and Identification

In the previous sections, we outlined the design of an unknown input observer that can be used to estimate the state of a dynamic system driven by partially unknown inputs. In the following sections we will essentially re-present Saif and Guan's (1993) approach to fault detection and identification using our modified UIO as the state estimator. First we need to establish a link between unknown input observers (UIO) and fault detection and identification (FDI). The design of the UIO in the previous sections was based on the assumption that a system model is known with precision. In reality, however, parameter values may be known only approximately or time-varying. There may be actuator failures and/or sensor failures which affect the behavior of the system. Let us now consider the effects of actuator faults, sensor faults, and parameter uncertainties on system dynamics and outputs one at a time. The only information commonly available about the faults is the location of their possible appearance. No assumption can be made about their mode, i.e., their time evolution and size. Suppose the nominal values of the parameters are known and our system can be linearized. The representation of the system can be written as:

$$\dot{x}_o = A_o x_o + B_o u_o \tag{3.6.1}$$

$$y_o = C_o x_o \tag{3.6.2}$$

where

subscript $_o$ means nominal system model parameters

$A_o \in \Re^{n \times n}$, $B_o \in \Re^{n \times q}$

$C_o \in \Re^{p \times n}$ is the measurement matrix

$x \in \Re^{n \times 1}$ is the state vector

$u \in \Re^{q \times 1}$ is the known or control input vector

$y \in \Re^{p \times 1}$ is the output vector

Actuator faults act directly on system dynamics. They affect the dynamic equations of the system model and these effects can be modeled using an actuator fault distribution matrix and an actuator fault vector in the following format:

$$\dot{x}_o = A_o x_o + B_o u_o + D_o d_o \tag{3.6.3}$$

where

$D_o \in \Re^{n \times m}$ is actuator fault distribution matrix

$d_o \in \Re^{m \times 1}$ is actuator fault vector

Sensor faults can be modeled as additive bias components in the output equation through a sensor fault distribution matrix.

$$y_o = C_o x_o + E_o e \tag{3.6.4}$$

where

$E_o \in \Re^{p \times r}$ is sensor fault distribution matrix

$e \in \Re^{r \times 1}$ is sensor fault vector

Parameter uncertainties may be initially modeled as deviations from their nominal values:

$$A = A_o + \Delta A \tag{3.6.5}$$

$$B = B_o + \Delta B \tag{3.6.6}$$

The uncertainty matrices ΔA and ΔB can be rewritten as

$$\Delta A = I_A \, \Delta A_a \tag{3.6.7}$$

$$\Delta B = I_B \, \Delta B_b \tag{3.6.8}$$

by giving the following definitions:

Definition 3.6.1 The n by k uncertainty indicator matrix I_R of any n by m matrix R is defined as $I_R(r_1, \ldots, r_k)$, where k is the number of rows of R that contain unknown elements. The jth column of this matrix has zero entries except for the a_j th entry which has a value of one.

As an example, if A is a 4 by 4 matrix and there are uncertain elements in the first and third row, then

$$k = 2, \quad a_1 = 1, \quad a_2 = 3, \text{ and } I_A(a_1, a_2) = I_A(1,3) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Definition 3.6.2 The k by m uncertainty matrix ΔR_r of any n by m matrix R is defined as

$$\Delta R_r = \begin{bmatrix} \Delta R_{r_1} \\ \vdots \\ \Delta R_{r_k} \end{bmatrix} \text{ where } \Delta R_{r_i} \text{ is the } r_i \text{ th row of } \Delta R$$

For example, in the above example of a 4 by 4 matrix A with uncertain elements in the first and third rows.

$$\Delta A_a = \begin{bmatrix} \Delta A_{a_1} \\ \Delta A_{a_2} \end{bmatrix} = \begin{bmatrix} \Delta A_1 \\ \Delta A_3 \end{bmatrix} = \begin{bmatrix} \Delta a_{11} & \Delta a_{12} & \Delta a_{13} & \Delta a_{14} \\ \Delta a_{31} & \Delta a_{32} & \Delta a_{33} & \Delta a_{34} \end{bmatrix}$$

Now the dynamic equations (3.6.1) can be rewritten to incorporate parameter uncertainties:

$$\dot{x} = (A_a + I_A \Delta A_a)\, x + (B_a + I_B \Delta B_b)\, u_a \qquad (3.6.9)$$

This can be further rewritten as

$$\dot{x} = A_a x + B_a u_a + I_A (\Delta A_a x) + I_B (\Delta B_b u_a) \qquad (3.6.10)$$

By defining

$$V = \begin{bmatrix} I_A & I_B \end{bmatrix} \qquad (3.6.11)$$

and

$$v = \begin{bmatrix} \Delta A_a x \\ \Delta B_b u_a \end{bmatrix} \qquad (3.6.12)$$

Parameter uncertainties can also be modeled as unknown inputs to a known system:

$$\dot{x} = A_a x + B_a u_a + V v \qquad (3.6.13)$$

So far in this section we just considered the individual effects of actuator faults, sensor faults, and parameter uncertainties one at a time. If any two or all of the three factors are present, then we can just stick the relevant terms into the dynamic equations and/or output equations. For example, in the simulation chapter we choose not to concern our selves with parameter variation of an UMS-2 robot and detect only two actuator faults and one sensor fault rather than component faults. In this case the model of the system is of the following form:

$$\dot{x}_a = A_a x_a + B_a u_a + D_a d_a \qquad (3.6.14)$$

$$y = C_a x_a + E_a e \qquad (3.6.15)$$

3.7 Actuator Fault Detection and Identification

As discussed in the previous section, actuator faults can be modeled as unknown inputs to a known system with known or nominal parameter values:

$$\dot{x}_o = A_o x_o + B_o u_o + D_o d_o \tag{3.7.1}$$

$$y_o = C_o x_o \tag{3.7.2}$$

where

subscript denotes original or open-loop systems

matrices and vectors are of appropriate orders as defined in section 3.6

Actuator fault vector d_o can be estimated by using the following theorem:

Theorem 3.7.1 The unknown input in system (3.7.1)-(3.7.2) can be estimated by an estimation technique of the following form if D_o is of full rank and T is a small enough sampling interval.

$$d_o(k) = (D_o^T D_o)^{-1} D_o^T (S(k) - B_o u_o(k)) \tag{3.7.3}$$

where

$$S(k) = A_o (e^{A_o kT} - I)^{-1} (x(k+1) - e^{A_o kT} x(k)) \tag{3.7.4}$$

and

$$d_o(k) = d_o(kT),\ S(k) = S(kT),\ x(k) = x(kT),\ u_o(k) = u_o(kT)$$

Proof: Applying the formula(Rugh, 1993) of the complete solution to a forced linear and continuous-time system in discrete form, the value of the state vector x(t) at time (k+1)T is:

$$x(k+1) = e^{A_o(k+1)T} (e^{-A_o kT} x(k) + \int_{kT}^{(k+1)T} (e^{-A_o \tau} (B_o u_o(k) + D_o d_o(k)) d\tau$$

$$= e^{A_o[(k+1)T]}\left(e^{-A_o(kT)}x(k) + (e^{-A_o(kT)} - e^{-A_o[(k+1)T]})A_o^{-1}(B_o u_o(k)+D_o d_o(k))\right)$$

$$= e^{A_o(kT)}x(k) + (e^{A_o(kT)} - I)\, A_o^{-1}(B_o u_o(k)+D_o d_o(k))$$

where T is the sampling period in the time domain

Defining S(k) as in (3.7.4) results in

$$B_o u_o(k)+D_o d_o(k) = S(k)$$

then

$$d_o(k) = (D_o^T D_o)^{-1} D_o^T (S(k) - B_o u_o(k))$$

It must be pointed out that the estimated state $\hat{x}(kT)$ rather than the real state x(kT) is to be used in evaluating S(k) in computer simulations. This is because the true state x(kT) is usually not available for measurement. It can only be asymptotically estimated by an estimator such as the UIO designed in the early sections of this chapter.

Plotting each component of $d_o(k)$ against time index k would show if the corresponding actuator has failed. This technique identifies not only the magnitude but also the shape of actuator faults

3.8 Sensor Fault Detection and Identification

Since no knowledge can be assumed about the time histories of sensor fault signals, it is reasonable to model them by a dynamic system driven by an unknown input signal. To do this we first present the following proposition:

Proposition 3.8.1 For any piecewise continuous vector function $f \in \Re^{r \times 1}$, and a stable $r \times r$ matrix A_f, there will always exist an input vector ξ such that

$$\dot{f} = A_f \, f + \xi \tag{3.8.1}$$

Proof: the proof is immediate simply by taking $\xi = \dot{f} - A_f \, f$.

Now we can assume that sensor faults have the following dynamics:

$$\dot{e} = A_e \, e + u_e \tag{3.8.2}$$

where

A_e	is a stable $r \times r$ matrix
u_e	is a $r \times 1$ unknown input vector
e	is a $r \times 1$ sensor fault vector

Augmenting (3.6.14)-(3.6.15) with (3.8.2) results in the following (n+r)th order dynamic system:

$$\begin{bmatrix} \dot{x}_o \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A_o & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} x_o \\ e \end{bmatrix} + \begin{bmatrix} B_o \\ 0 \end{bmatrix} u + \begin{bmatrix} D_o & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} d_o \\ u_e \end{bmatrix} \tag{3.8.3}$$

$$y_o = \begin{bmatrix} C & E \end{bmatrix} \begin{bmatrix} x_o \\ e \end{bmatrix} \tag{3.8.4}$$

Define

$$x = \begin{bmatrix} x_o \\ e \end{bmatrix}$$

$$A = \begin{bmatrix} A_o & 0 \\ 0 & A_e \end{bmatrix}$$

$$B = \begin{bmatrix} B_o \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} C & E \end{bmatrix}$$

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

$$d = \begin{bmatrix} d_o \\ u_c \end{bmatrix}$$

then (3.8.3)-(3.8.4) is in the standard form of (3.2.1)-(3.2.2) of section 3.2:

$$\dot{x} = Ax + Bu + Dd \tag{3.8.5}$$

$$y = Cx \tag{3.8.6}$$

Therefore the state of the system in (3.8.5)-(3.8.6) can be estimated by using the UIO designed in section 3.4 provided that all the necessary and sufficient conditions related to the existence of an UIO are satisfied.

It is now clear that sensor faults are part of the state of the augmented system. Therefore monitoring the state estimates ($\begin{bmatrix} \hat{x}_o \\ \hat{e} \end{bmatrix}$) would provide an immediate means of the detection of sensor failures. The failure detection logic is very simple. Any nonzero component of $\hat{e}$ would indicate a sensor failure. It is also extremely easy to identify or isolate sensor failure(s) by checking which component of $\hat{e}$ is nonzero. For example, if only the second component of $\hat{e}$ is nonzero, then only sensor 2 has failed; if the first two components of $\hat{e}$ are nonzero, then both sensor 1 and sensor 2 have failed.

Plotting each component of $\hat{e}$ would indicate if the corresponding sensor has failed. This technique could identify not only the magnitude but also the shape of sensor faults.

3.9 Summary

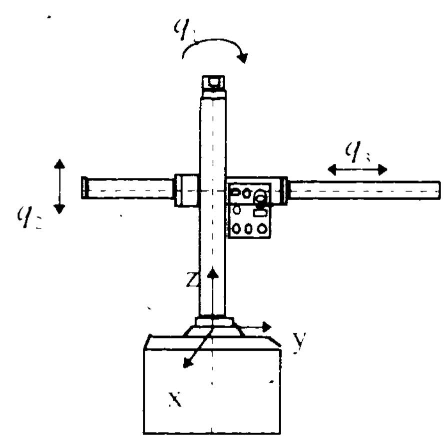
In the first five sections of this chapter we presented a modified unknown input observer (UIO) capable of estimating the state of a linear dynamic system driven by both known and unknown inputs. By performing a couple of similarity transformations and a nonsingular transformation, we were able to partition the system into three subsystems. One of these subsystems was a dynamic system driven by known inputs only. The other two subsystems are nothing but explicit algebraic relationships between the states of the subsystems and the measurable outputs. This made it possible to use a conventional Luenberger observer with slight modifications to estimate the state of the transformed system. The estimate of the state of the original system can be obtained by performing inverse transformations. It was possible to state a similar necessary and sufficient condition to that of a conventional observer for the existence of a stable estimator and arbitrary pole placement capability. It was also shown and proven that this necessary and sufficient condition can be expressed in terms of original system matrices. This alternative expression of the necessary and sufficient condition provides a much easier way for checking whether the condition is satisfied before any transformations are undertaken. In view of a couple of competing UIO design methods, it is felt that the design and computational complexities involved in designing UIO's is greatly reduced in our proposed approach. Our simulation program in Appendix C also shows that our UIO algorithm is quite easy to code.

In the last few sections of this chapter we used our modified UIO in fault detection and identification of uncertain dynamic systems. We were able to model parameter uncertainties as unknown inputs to a known system with nominal or assumed parameter values. We also modeled actuator faults as unknown inputs to the dynamic equations of a known system because they act directly onto system dynamics. We dealt with sensor faults by modeling them as additive biases to the output equations. We used a generalized inverse solution technique in estimating the actuator fault vector for the purpose of actuator fault detection and identification. This technique can also be used to estimate one or more parameter variations in some systems. By modeling sensor faults as the state of a dynamic system driven by unknown input, we were able to obtain an augmented system whose state vector contains not only the original state variables but also sensor fault signals. We could thus obtain the estimates of sensor faults by extracting a sub-vector from the estimate of the state vector of the augmented system. The estimates of sensor faults provides an immediate means of sensor fault detection and identification (FDI). In both actuator FDI and sensor FDI, we were able to obtain not only the shape but also the magnitude of the faults This enables us to distinguish between a momentary fault that clears its self and a persistent one. It is recognized that this UIO based FDI approach allows us to detect and identify multiple and/or even simultaneous actuator and sensor faults as well as parameter variations so long as the total number of faults and uncertainties to be detected and identified is less than the number of available outputs.

Chapter 4 FDI Study of a Constrained Mechanical System - Approach and Simulation

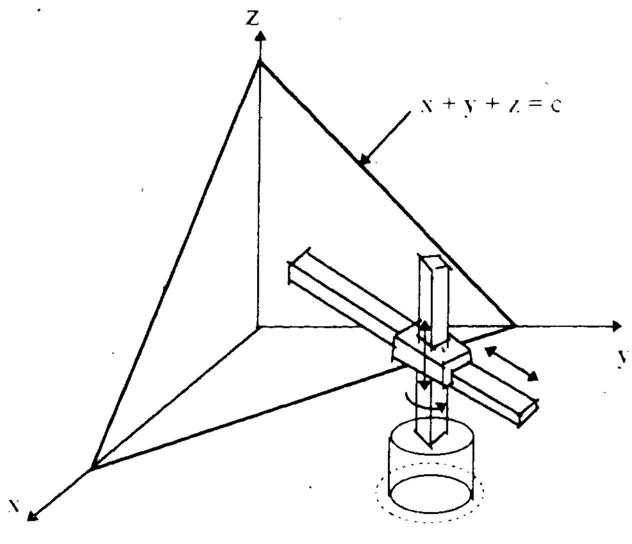
4.1 Introduction

In chapter 2, constrained mechanical systems were initially mathematically described by nonlinear equations with Lagrange multipliers. Linearization was performed using standard Taylor series expansion. The peculiar structure and important properties of linearized constrained mechanical systems were analyzed and normal (dynamic) forms of the system representations were derived. The resultant purely dynamic subsystem of the linear mechanical descriptor system representation is in the form of linear time-invariant dynamic equations. This allows us to shift our analysis from the domain of linear singular system theory to the domain of linear system theory. In chapter 3, an observer design method was proposed for linear time-invariant dynamic systems driven by both known and unknown inputs and a FDI approach based on UIO theory was presented. We were able to model actuator faults as unknown inputs and sensor faults as additive biases to the outputs. In this chapter, we combine the results of chapters 2 and 3 and use them in fault detection and control of a UMS-2 robot manipulator system. The following is a drawing of this robot.



UMS-2 robot

This robot has three degrees of freedom during unconstrained motion. However, we study it in the context of motion with a holonomic constraint. A sketch of the robot manipulator geometric workspace is given in the following.



UMS-2 Robot Manipulator Task Geometry

This robot was found in the paper of Mills & Goldenberg (1989) This paper used this robot as a numerical example in force and position control of manipulators during constrained motion tasks It gave little information on the nature of the specific tasks performed by this robot manipulator For the purpose of simulation it is sufficient to know that the UMS-2 robot is assumed to be in contact with a rigid frictionless surface Robots which are similar to but not identical with a UMS-2 robot can be found in the book of Vukobratovic & Potkonjak (1982) Some of these robot manipulators can perform tasks such as spraying powder along a prescribed trajectory

4.2 Approach and Simulation

We deploy a systematic approach to fault detection and identification of the UMS-2 robot manipulator system This approach may be outlined as follows:

1) Write original nonlinear mathematical description of the system with actuator faults appended to dynamic equations and sensor faults appended to output equations
2) Linearize the nonlinear model and rewrite it in a generalized state space format
3) Perform a nonsingular coordinate transformation and derive the normal form
4) Perform similarity transformations to bring the dynamic subsystem into its canonical form

4) Design an unknown input observer and a state-feedback controller for the dynamic subsystem

5) Obtain necessary results and convert the results back into the original coordinates by reversing transformations

In the simulations we will detect and identify two actuator faults and one sensor fault. The dynamic equations of motion of this manipulator in unconstrained form are given by:

$$M_i(q)\ \ddot{q}_i + H_i(q,\dot{q}) = T_i + F_i \qquad (4.2.1)$$

where

$$M_1(q) = J_{z1} + J_{z2} + J_{z3} + m_3(q_3 + l_3)^2$$

$$H_1(q,\dot{q}) = 2m_3(q_3 + l_3)\dot{q}_1\dot{q}_3$$

$$M_2(q) = m_2 + m_3$$

$$H_2(q,\dot{q}) = (m_2 + m_3)g$$

$$M_3(q) = m_3$$

$$H_3(q,\dot{q}) = -m_3(q_3 + l_3)\dot{q}_1^2$$

T_i, $i = 1, 2, 3$ are known inputs that represent control signals

F_i, $i = 1, 2, 3$ are unknown inputs that represent actuator faults or failures

The output equations chosen for the simulations are as follows

$$r = \psi(q,\dot{q}) = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 + \dot{q}_2 + \dot{q}_3 \\ \dot{q} \end{bmatrix} \qquad (4.2.2a)$$

$$r_f = r + \mathrm{E}\,\mathrm{e} \qquad (4.2.2b)$$

where

$$E = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ is sensor fault distribution vector}$$

e is an unknown signal representing a sensor fault or failure

r_d is output vector with appended sensor fault

The position vector p is given by

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (q_3 + l_3)\cos q_1 \\ (q_3 + l_3)\sin q_1 \\ q_2 \end{bmatrix}$$

The constraint function representing the robot end-effector being in contact with a rigid flat surface is the following:

$$\phi(p) = x + y + z - c = 0$$

This holonomic constraint equation can be rewritten as

$$\theta(q) = (q_3 + l_3)\cos q_1 + (q_3 + l_3)\sin q_1 + q_2 - c = 0 \qquad (4.2.3)$$

where

$$c = x_d + y_d + z_d$$

$$x_d = (q_{3d} + l_3)\cos q_{1d}$$

$$y_d = (q_{3d} + l_3)\sin q_{1d}$$

$$z_d = q_{2d}$$

We linearize this constrained mechanical system about a point at which the robot manipulator is stationary but being in contact with a flat surface. The nominal dynamic parameters and the nominal values of the generalized state variables are given in the following table:

NOMINAL DYNAMIC AND KINEMATIC PARAMETERS

Nominal State	Nominal Dynamic Parameters
$q_{1o} = 0.4363$ radian	$l_3 = 0.2$ m
$q_{2o} = 0.3$ m	$m_2 = 1$ kg
$q_{3o} = 0.3$ m	$m_3 = 2$ kg
$\Lambda_o = \frac{-1}{\sqrt{3}} \frac{kg-m}{s^2}$	$J_{z1} = 0.1\ kg-m^2$
$\dot{q}_{1o} = 0$	$J_{z2} = 0.2\ kg-m^2$
$\dot{q}_{2o} = 0$	$J_{z3} = 0.1\ kg-m^2$
$\dot{q}_{3o} = 0$	
$\ddot{q}_{1o} = 0$	
$\ddot{q}_{2o} = 0$	
$\ddot{q}_{3o} = 0$	

The above dynamic system with constraint but without sensor fault can be described by the following equations

$$M(q)\ddot{q} + H(q,\dot{q}) = J^T\Lambda + T + F \tag{4.2.4}$$

$$r = \psi(q,\dot{q}) \tag{4.2.5}$$

where

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$M(q)=\begin{bmatrix} M_1(q) & 0 & 0 \\ 0 & M_2(q) & 0 \\ 0 & 0 & M_3(q) \end{bmatrix}$$

$$H(q,\dot{q})=\begin{bmatrix} H_1(q,\dot{q}) \\ H_2(q,\dot{q}) \\ H_3(q,\dot{q}) \end{bmatrix}$$

$$T=\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

$$F=\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

$$r=\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

The Jacobian in this single holonomic constraint system may be defined as

$$J(q)=\frac{\partial\theta(q)}{\partial q}$$

Now applying the linearization approach outlined in Section 2.2 of Chapter 2 results in:

$$M=M(q_0(t))=\begin{bmatrix} J_{z1}+J_{z2}+J_{z3}+m_3(q_{3_0}+l_3)^2 & 0 & 0 \\ 0 & m_2+m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$\Rightarrow \qquad M=\begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$K = \frac{\partial H(q,\dot{q})}{\partial q}\bigg|_{(q_0,\dot{q}_0)} - \Lambda_0 \frac{\partial J^T(q)}{\partial q}\bigg|_{(q_0,\dot{q}_0)}$$

$$= \begin{bmatrix} 0 & 0 & 2m_3 \\ 0 & 0 & 0 \\ 0 & 0 & -m_3 \end{bmatrix} - \left(\frac{-1}{\sqrt{3}}\right) \frac{\partial}{\partial q}\left(\frac{\partial \theta}{\partial q}\right)\bigg|_{(q_0,\dot{q}_0)}$$

$$\Rightarrow \qquad K = \begin{bmatrix} 0 & 0 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix} + \begin{bmatrix} 0.3836 & 0 & -0.2793 \\ 0 & 0 & 0 \\ -0.2793 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.3836 & 0 & 3.7207 \\ 0 & 0 & 0 \\ -0.2793 & 0 & -2 \end{bmatrix}$$

$$D = \frac{\partial H(q,\dot{q})}{\partial \dot{q}}\bigg|_{(q_0,\dot{q}_0)} = \begin{bmatrix} \frac{\partial H_1(q,\dot{q})}{\partial \dot{q}_1} & \frac{\partial H_1(q,\dot{q})}{\partial \dot{q}_2} & \frac{\partial H_1(q,\dot{q})}{\partial \dot{q}_3} \\ \frac{\partial H_2(q,\dot{q})}{\partial \dot{q}_1} & \frac{\partial H_2(q,\dot{q})}{\partial \dot{q}_2} & \frac{\partial H_2(q,\dot{q})}{\partial \dot{q}_3} \\ \frac{\partial H_3(q,\dot{q})}{\partial \dot{q}_1} & \frac{\partial H_3(q,\dot{q})}{\partial \dot{q}_2} & \frac{\partial H_3(q,\dot{q})}{\partial \dot{q}_3} \end{bmatrix}_{(q_0,\dot{q}_0)}$$

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

$$L = \frac{\partial \theta(q)}{\partial q}\bigg|_{(q_0,\dot{q}_0)} = \begin{bmatrix} \frac{\partial \theta(q)}{\partial q_1} & \frac{\partial \theta(q)}{\partial q_2} & \frac{\partial \theta(q)}{\partial q_3} \end{bmatrix}_{(q_0,\dot{q}_0)}$$

$$\Rightarrow \qquad L = \begin{bmatrix} 0.2418 & 1 & 1.3289 \end{bmatrix}$$

$$C_f = \frac{\partial \psi(q,\dot{q})}{\partial q}\bigg|_{(q_0,\dot{q}_0)}$$

$$\Rightarrow \qquad C_f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_v = \frac{\partial\psi(q,\dot{q})}{\partial\dot{q}}\bigg|_{(q_0,\dot{q}_0)}$$

$$\Rightarrow \quad C_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

The Jacobian J of the system is of the form: $J = L = \begin{bmatrix} 0.2418 & 1 & 1.3289 \end{bmatrix}$

The linearized descriptor form representation of the system is

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & J^T \\ L & 0 & 0 \end{bmatrix}\begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} \tag{4.2.6}$$

$$y_t = C_p\, z + C_v\, \dot{z} \tag{4.2.7}$$

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -0.3836 & 0 & -3.7207 & 0 & 0 & 0 & 0.1935 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.2793 & 0 & 2 & 0 & 0 & 0 & 1.3289 \\ 0.2418 & 1 & 1.3289 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \lambda \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{4.2.8}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} \tag{4.2.9}$$

where

$z = q - q_0$ is the generalized coordinates in linearized form

$\dot{z} = \dot{q} - \dot{q}_0$ is the derivative of generalized coordinates in linearized form

$\lambda = \Lambda - \Lambda_0$ is the constrained force in linearized form

$f = T - T_0$ is the known input signal in linearized form

$d = F - F_0$ is the actuator fault or failure signal in linearized form

$y = r - r_0$ is the output vector in linearized form

$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$ is the partition of the displacement vector

$f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$ is the partition of the known input vector

$d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$ is the partition of the unknown input vector

Now we can apply the nonsingular transformation technique presented in section 2.3 of chapter 2 to derive the normal form of the linearized descriptor form representation.

It can be verified that the following nonsingular transformation matrix

$$T = \begin{bmatrix} 0 & 4.1356 & 0 \\ 1 & -1 & 1 \\ -0.7525 & 0 & 0 \end{bmatrix} \tag{4.2.10}$$

satisfies $LT = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

Define $$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = T^T z \qquad (4.2.11)$$

where

$$\xi_1 \in \Re^{(n-q)}, \quad \xi_2 \in \Re^q, \quad n = 3, \quad q = 1$$

Pre-multiplying both sides of (4.2.6) by $\begin{bmatrix} T^T & 0 & 0 \\ 0 & T^T & 0 \\ 0 & 0 & I_q \end{bmatrix}$ and performing lengthy simplification will yield the following normal form representation:

$$\begin{bmatrix} \dot{\xi}_1 \\ \ddot{\xi}_1 \end{bmatrix} = \begin{bmatrix} 0 & I_{n-q} \\ -M_s^{-1}K_s & -M_s^{-1}D_s \end{bmatrix} \begin{bmatrix} \xi_1 \\ \dot{\xi}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ M_s^{-1} \end{bmatrix} \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ M_s^{-1} \end{bmatrix} \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} \qquad (4.2.12)$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \dot{\xi}_1 \end{bmatrix} \qquad (4.2.13)$$

where

$$M_s = M_{11} - M_{12} \frac{L_{11}}{L_{12}} \qquad (4.2.14)$$

$$K_s = K_{11} - K_{12} \frac{L_{11}}{L_{12}} \qquad (4.2.15)$$

$$D_s = D_{11} - D_{12} \frac{L_{11}}{L_{12}} \qquad (4.2.16)$$

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = T^T M (T^T)^{-1} \qquad (4.2.17)$$

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} = T^T K (T^T)^{-1} \qquad (4.2.18)$$

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = T^T D (T^T)^{-1} \qquad (4.2.19)$$

$$\begin{bmatrix} L_{11} & L_{12} \end{bmatrix} = L (T^T)^{-1} \qquad (4.2.20)$$

$$C_1 = C_p (T^T)^{-1} \begin{bmatrix} I_{n-q} \\ -\dfrac{L_{11}}{L_{12}} \end{bmatrix} \qquad (4.2.21)$$

$$C_2 = C_v (T^T)^{-1} \begin{bmatrix} I_{n-q} \\ -\dfrac{L_{11}}{L_{12}} \end{bmatrix} \qquad (4.2.22)$$

$$\begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \bar{f}_3 \end{bmatrix} = T^T \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \qquad (4.2.23)$$

$$\begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \bar{d}_3 \end{bmatrix} = T^T \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \qquad (4.2.24)$$

The algebraic part of the system is described by the following:

$$\xi_2 = -\frac{L_{11}}{L_{12}} \xi_1 \qquad (4.2.25)$$

$$\begin{aligned} \lambda = \Big\{ & \left(\begin{bmatrix} D_{21} & M_{21} \end{bmatrix} + \frac{L_{11}}{L_{12}} \begin{bmatrix} D_{22} & M_{22} \end{bmatrix} \right) \begin{bmatrix} 0 & I_{n-q} \\ -M_s^{-1} K_s & -M_s^{-1} D_s \end{bmatrix} \\ & + \left(\begin{bmatrix} K_{21} & 0 \end{bmatrix} + \frac{L_{11}}{L_{12}} \begin{bmatrix} K_{22} & 0 \end{bmatrix} \right) \Big\} \begin{bmatrix} \xi_1 \\ \dot{\xi}_1 \end{bmatrix} \\ & + \left(\begin{bmatrix} D_{21} & M_{21} \end{bmatrix} + \frac{L_{11}}{L_{12}} \begin{bmatrix} D_{22} & M_{22} \end{bmatrix} \right) \begin{bmatrix} 0 \\ M_s^{-1} \end{bmatrix} \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix} - \bar{f}_3 - \bar{d}_3 \end{aligned} \qquad (4.2.26)$$

Evaluation of equations (4.2.12)-(4.2.13) results in the following numerical form:

$$\begin{bmatrix} \dot{\xi}_1 \\ \ddot{\xi}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.3392 & -0.0028 & 0 & 0 \\ 8.3400 & 0.0466 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \dot{\xi}_1 \end{bmatrix}$$

Evaluation of equations (4.2.12)-(4.2.13) results in the following numerical form:

$$\begin{bmatrix}\dot{\xi}_1\\ \dot{\xi}_2\\ \ddot{\xi}_1\\ \ddot{\xi}_2\end{bmatrix}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0.3392 & -0.0028 & 0 & 0\\ 8.3400 & 0.0466 & 0 & 0\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2\\ \dot{\xi}_1\\ \dot{\xi}_2\end{bmatrix}$$

$$+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0.3851 & 0.0085\\ 0.5360 & 1.0717\end{bmatrix}\begin{bmatrix}\bar{f}_1\\ \bar{f}_2\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0.3851 & 0.0085\\ 0.5360 & 1.0717\end{bmatrix}\begin{bmatrix}\bar{d}_1\\ \bar{d}_2\end{bmatrix} \tag{4.2.27}$$

$$y=\begin{bmatrix}-0.1512 & 0.2468 & 0 & 0\\ -0.6252 & 0.0207 & 0 & 0\\ -2.1598 & 0.0275 & -2.7850 & 0.0482\\ 0 & 0 & -0.1512 & 0.2468\end{bmatrix}\begin{bmatrix}\xi_1\\ \xi_2\\ \dot{\xi}_1\\ \dot{\xi}_2\end{bmatrix} \tag{4.2.28}$$

The above representation can be rewritten in the following unaugmented open-loop form

$$\dot{x}_u = A_u x_u + B_u u_u + D_u d_u \tag{4.2.29}$$

$$y_u = C_u x_u \tag{4.2.30}$$

where

$$x_u = \begin{bmatrix}\xi\\ \dot{\xi}\end{bmatrix} \tag{4.2.31}$$

$$u_u = \begin{bmatrix}\bar{f}_1\\ \bar{f}_2\end{bmatrix} \tag{4.2.32}$$

$$d_u = \begin{bmatrix}\bar{d}_1\\ \bar{d}_2\end{bmatrix} \tag{4.2.33}$$

$$A_u = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0.3392 & -0.0028 & 0 & 0\\ 8.3400 & 0.0466 & 0 & 0\end{bmatrix} \tag{4.2.34}$$

$$B_o = D_o = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.3851 & 0.0085 \\ 0.5360 & 1.0717 \end{bmatrix} \tag{4.2.35}$$

$$C_o = \begin{bmatrix} -0.1512 & 0.2468 & 0 & 0 \\ -0.6252 & 0.0207 & 0 & 0 \\ -2.1598 & 0.0275 & -2.7850 & 0.0482 \\ 0 & 0 & -0.1512 & 0.2468 \end{bmatrix} \tag{4.2.36}$$

The open-loop system is not stable because A_o has eigenvalues ± 0.5235 and $\pm 0.6675i$. Hence state feedback is used to stabilize the open-loop system. The closed-loop system takes the form

$$\begin{aligned} \dot{x}_c &= A_o x_c + B_o u_o + D_o d_o - B_o K_c \hat{x}_c \\ &= A_c x_c + B_o u_o + D_o d_o \end{aligned} \tag{4.2.37}$$

$$y_c = C_o x_c \tag{4.2.38}$$

where

subscript c denotes closed-loop and $\hat{x}_c$ is an estimate of x_c

By placing the poles of the closed-loop matrix

$$A_c = A_o - B_o K_c \tag{4.2.39}$$

at arbitrarily chosen locations $-5 \pm 4i$ and $-6 \pm 2i$, the state feedback gain matrix K_c and the matrix A_c are computed by MATLAB as

$$K_c = \begin{bmatrix} 108.3527 & -0.8201 & 26.2529 & -0.2808 \\ -46.4048 & 37.7784 & -13.1420 & 11.3378 \end{bmatrix}$$

$$A_c = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -41.0000 & -0.0062 & -10.0001 & 0.0124 \\ -0.0062 & -40.0000 & 0.0124 & -11.9999 \end{bmatrix}$$

As discussed in chapter 5, sensor fault estimation is accommodated by adding a term Ee in the output equations of all previous representations, i.e.,

$$y_a = C_o x_c + Ee \tag{4.2.40}$$

The augmented open-loop system is of the following form:

$$\begin{bmatrix} \dot{x}_o \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A_o & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} x_o \\ e \end{bmatrix} + \begin{bmatrix} B_o \\ 0 \end{bmatrix} u + \begin{bmatrix} D_o & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} d_o \\ u_e \end{bmatrix} \tag{4.2.41}$$

$$y_a = \begin{bmatrix} C_o & E \end{bmatrix} \begin{bmatrix} x_o \\ e \end{bmatrix} \tag{4.2.42}$$

where

A_e is a stability matrix(a negative constant in one dimensional case)

u_e is a sensor fault input vector(a scalar function in the above case)

Define

$$x_a = \begin{bmatrix} x_o \\ e \end{bmatrix}$$

$$A_a = \begin{bmatrix} A_o & 0 \\ 0 & A_e \end{bmatrix}$$

$$B_a = \begin{bmatrix} B_o \\ 0 \end{bmatrix}$$

$$C_a = \begin{bmatrix} C_o & E \end{bmatrix}$$

$$D_a = \begin{bmatrix} D_o & 0 \\ 0 & I \end{bmatrix}$$

$$d_a = \begin{bmatrix} d_o \\ u_e \end{bmatrix}$$

Then last representation (4.2.41)-(4.2.42) can be written as

$$\dot{x}_a = A_a x_a + B_a u_o + D_a d_a \tag{4.2.43}$$

$$y_a = C_a x_a \tag{4.2.44}$$

Once again state feedback is used to stabilize the above system. The augmented closed-loop dynamic system representation is

$$\dot{x}_{ac} = A_{ac} x_{ac} + B_a u_o + D_a d_a \tag{4.2.45}$$

$$y = C_a x_{ac} \tag{4.2.46}$$

where

$$A_{ac} = A_a - B_a K_{ac} \tag{4.2.47}$$

$$x_{ac} = \begin{bmatrix} x_c \\ e \end{bmatrix} \tag{4.2.48}$$

Let us now determine K_{ac} and A_{ac} by performing the following analysis:

Assuming K_{ac} is of the form $[\, K_c \;\; K_{arb} \,]$ where K_c is the state feedback gain matrix used in the unaugmented case and K_{arb} is an unknown submatrix to be determined.

$$A_{ac} = A_a - B_a K_{ac} = \begin{bmatrix} A_o & 0 \\ 0 & A_e \end{bmatrix} - \begin{bmatrix} B_o \\ 0 \end{bmatrix} [\, K_c \;\; K_{arb} \,]$$

$$= \begin{bmatrix} A_o - B_o K_c & -B_o K_{arb} \\ 0 & A_e \end{bmatrix}$$

$$= \begin{bmatrix} A_c & -B_o K_{arb} \\ 0 & A_e \end{bmatrix}$$

where

A_c is the unaugmented closed-loop matrix defined previously

Note that

$\det(A_{ac}) = \det(A_c) \times \det(A_e)$

poles of A_{ac} are poles of A_c plus pole(s) of A_e

Therefore no matter what value K_{arb} takes the eigenvalues of A_{ac} remain the same. In this simulation we use the following arbitrary values:

$$K_{arb} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and $$A_e = -5$$

then the resultant A_{ac} computed by MATLAB using the aforementioned expression is

$$A_{ac} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ -41.0000 & -0.0062 & -10.0001 & 0.0124 & -0.3936 \\ -0.0062 & -40.0000 & 0.0124 & -11.9999 & 1.6077 \\ 0 & 0 & 0 & 0 & -5.0000 \end{bmatrix}$$

with poles or eigenvalues at

-5.0000 + 4.0000 i

-5.0000 - 4.0000 i

-6.0000 + 2.0000 i

-6.0000 - 2.0000 i

-5.0000

The estimation of the state vector of a system that has actuator faults and/or sensor faults relies on the evaluation of an unknown input observer(UIO) outlined in chapter 3. In the MATLAB simulation program, the following were done or obtained:

1. Necessary and sufficient conditions for the existence of an UIO are verified numerically
2. Two similarity transformations are performed in bring the augmented closed-loop linear

dynamic system to its special canonical form and partitioning it into three subsystems

3, The output vector Y_c of the unaugmented closed-loop system is obtained by doing a linear dynamic system simulation using the *lsim* command in MATLAB

4, The output vector Y_{ac} of the augmented closed-loop system is calculated using equation

$$Y_{ac} = Y_c + \mathrm{E\,e}$$

where

$$\mathrm{E} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$ is sensor fault distribution matrix (vector)

$\mathrm{e} = u_c = 0.5*\mathrm{u(t-3)}$ is the assumed form of sensor 1 failure

5, The observer equation is: $\dot{w} = -6\,\mathrm{w} + \begin{bmatrix} 0 & -1.0054 & 0.0330 & 0.0050 \end{bmatrix} Y_{ac} + \begin{bmatrix} 0 & 0 \end{bmatrix} \mathrm{u}_a$

6, The state vector of the augmented closed-loop system is estimated using the equation:

$$\hat{x}_{ac} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathrm{w} + \begin{bmatrix} 1 & 148.3324 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} Y_{ac} \qquad (4.2.49)$$

7, The estimate of the state vector of the unaugmented closed-loop system $\hat{x}_c$ is extracted from $\hat{x}_{ac}$ using equation $\hat{x}_{ac} = \begin{bmatrix} \hat{x}_c \\ \hat{e} \end{bmatrix}$

8, The estimate of sensor fault $\hat{e}$ is also extracted from $\hat{x}_{ac}$ using equation $\hat{x}_{ac} = \begin{bmatrix} \hat{x}_c \\ \hat{e} \end{bmatrix}$

Once the estimate of the state vector of the unaugmented closed-loop system $\hat{x}_c$ is available, transformed actuator fault vector $\mathrm{d}_a = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix}$ can be estimated using the least square solution technique presented in chapter 3, i.e.,

$$\begin{bmatrix} \hat{\bar{d}}_1(kT) \\ \hat{\bar{d}}_2(kT) \end{bmatrix} = \hat{d}_o\,(k) = (D_o^T D_o)^{-1} D_o^T\ (S(k) - B_o\, u_o\,(k)) \tag{4.2.50}$$

where

$$S(k) = A_c\ (e^{A_c T}\ -I)^{-1}(\hat{x}_c\ (k+1) - e^{A_c T}\ (k))$$

$$S(k)= S(kT),\ \hat{x}_c(k)= \hat{x}_c(kT),\ u_o\,(k)= u_o\,(kT)$$

and

$\hat{\bar{d}}_1$ is the estimate of transformed actuator fault $\bar{d}_1$

$\hat{\bar{d}}_2$ is the estimate of transformed actuator fault $\bar{d}_2$

$\hat{x}_c$ is the estimate of the state vector of the unaugmented closed-loop system

Then estimates of the original actuator faults can be obtained by reversing the transformation defined by equation (4.2.24), i.e.,

$$\begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \\ \hat{\bar{d}}_3 \end{bmatrix} = T^T \begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \\ \hat{d}_3 \end{bmatrix} \tag{4.2.51a}$$

$$\Rightarrow \quad \begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \\ \hat{d}_3 \end{bmatrix} = (T^T)^{-1} \begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \\ \hat{\bar{d}}_3 \end{bmatrix} \tag{4.2.51b}$$

Note that the algebraic equation (4.2.26) can not be used to estimate transformed actuator fault $\bar{d}_3$ because the constrained force λ in this equation is also unknown. This means that $\hat{\bar{d}}_3$ in equation (4.2.51) can not be determined or evaluated. The variables that we do have estimates for are just $\bar{d}_1$ and $\bar{d}_2$. To obtain the estimates of d_1, d_2, and d_3 from the estimates of $\bar{d}_1$ and $\bar{d}_2$ is equivalent to solving a system of two linear equations with three unknowns as specified by (4.2.51). A solution can only be obtained by assuming one of the estimates of d_1, d_2, and d_3 is zero or known. This essentially requires that one of the actuators is faultless. In this particular system it does not matter which actuator is assumed to be healthy. As long as one of the three actuators can be assumed faultless, the other two actuator faults can be uniquely detected and identified by solving a system of two linear equations with two unknowns. For example, suppose the 3-rd actuator is faultless, i.e., $d_3 = 0 = \hat{d}_3$, then $\hat{d}_1$ and $\hat{d}_2$ can be determined by solving two equations. $\hat{\bar{d}}_3$ can be obtained via the transformation and used in (4.2.26) to generate an estimate of the constraint force λ.

In this thesis we performed two simulations to estimate all three actuator faults. The first simulation estimates actuator 1 and actuator 2 faults based on the assumption that actuator 3 is highly reliable and faultless. The second simulation estimates actuator 1 and actuator 3 faults based on the assumption that actuator 2 is highly reliable and faultless.

Although the unknown inputs representing two actuator failures in the simulations can be of any form, we have to specify a specific function for each one of them for the purpose of estimation. In this simulation we just happen to use step function as a form of possible failures. The soft actuator failures are assumed to be of the following form:

(a) For simulation number 1

$$d_1 = 0.5*u(t-3)$$

$$d_2 = 0.4*u(t-3)$$

$$d_3 = 0.0*u(t-3)$$

(b) For simulation number 2

$$d_1 = 0.5*u(t-3)$$

$$d_2 = 0.0*u(t-3)$$

$$d_3 = 0.4*u(t-3)$$

The known control inputs are assumed to be of the following form for both simulations:

$$f_1 = u_1 = 7*u(t)$$

$$f_2 = u_2 = 8*u(t)$$

$f_3 = u_3 = 9 * \mathrm{u(t)}$

Given the above control inputs f_1, f_2, & f_3 and actuator fault inputs d_1, d_2, & d_3, we can use equations (4.2.23)-(4.2.24) and equations (4.2.32)-(4.2.33) to obtain the known inputs and unknown inputs used for simulating the normal form (4.2.37)

(a) for simulation number 1

$$\mathrm{u}_n = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -0.7525 \\ 4.1356 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (4.2.52)$$

$$\mathrm{d}_n = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4.1356 & -1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (4.2.53)$$

where

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} = T^T = \begin{bmatrix} 0 & 1 & -0.7525 \\ 4.1356 & -1 & 0 \\ 0 & 1 & -0 \end{bmatrix} \quad (4.2.54)$$

(b) for simulation number 2

$$\mathrm{u}_n = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -0.7525 \\ 4.1356 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad (4.2.55)$$

$$\mathrm{d}_n = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{13} \\ T_{21} & T_{23} \end{bmatrix} \begin{bmatrix} d_1 \\ d_3 \end{bmatrix} = \begin{bmatrix} 0 & -0.7525 \\ 4.1356 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_3 \end{bmatrix} \quad (4.2.56)$$

Then the original actuator fault estimates are obtained by performing reverse transformation on the estimates generated by the simulations. The reverse transformation equation (4.2.51b) reduces to the following forms for the following cases:

(a) for simulation number 1

$$\begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \end{bmatrix} = \begin{bmatrix} 0.2418 & 0.2418 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \end{bmatrix} \quad (4.2.57)$$

(b) for simulation number 2

$$\begin{bmatrix} \hat{d}_1 \\ \hat{d}_3 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{13} \\ T_{21} & T_{23} \end{bmatrix}^{-1} \begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \end{bmatrix} \doteq \begin{bmatrix} 0 & 0.2418 \\ -1.3289 & 0 \end{bmatrix} \begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \end{bmatrix} \tag{4.2.58}$$

where

$\begin{bmatrix} \hat{\bar{d}}_1 \\ \hat{\bar{d}}_2 \end{bmatrix}$ was obtained in the simulations using equation (4.2.50)

Once the estimates of the original actuator faults are obtained, the estimates and their corresponding fault signals are plotted for easy fault detection and identification. In simulation number 1, $\hat{d}_1$ (the estimate of d_1) and d_1 itself are plotted against time to show the transient and asymptotic behavior of actuator 1 fault. $\hat{d}_2$ (the estimate of d_2) and d_2 itself are plotted against time to show the transient and asymptotic behavior of actuator 2 fault. Each of the two plots shows that the estimate has a big spike initially, another spike at the 3-rd second, and then quickly settles down to the asymptotic value. The first spike is due to the transient behavior. The second spike indicates that the actuator had a fault at time t = 3 seconds. The asymptotic behavior confirms the stability of the observer and the correctness of the theoretical work. Similarly in simulation number 2, actuator 1 and actuator 3 faults are detected and identified using two similar plots. The combination of simulation number 1 and simulation number 2 detects and identifies all three(3) actuator faults.

It can be seen from equation (4.2.57) that sensor fault is part of the state of the augmented system. Sensor fault estimate can be obtained from the estimate of the state vector of the augmented system. Sensor fault estimate provides an immediate means for sensor fault detection and identification.

The plot of sensor fault estimate $\hat{e}$ and the original assumed sensor fault function shows a sensor fault at time t = 3 seconds. The objective of sensor fault detection is achieved.

In this particular simulation case, we performed two simulations each of which is based on the assumption that one of the three actuators is highly reliable(faultless). The observer equations for these two simulations are essentially the same. Consequently, we could say that we used only one observer (but two simulations). In fact, it can be seen from Appendix C that the first part of the source codes of the two simulations are identical. The difference only exists in the last part of the program. This is the reason that two simulations were written in one source code program.

The plots of actuator and sensor faults and their corresponding estimates against time are shown in the figures of this thesis.

4.3 Summary

In conclusion, this chapter has illustrated a systematic or at least a procedural approach to fault detection and identification of a major subclass of generalized state-space systems. By performing several nonsingular and similarity transformations and using an unknown input observer we were able to convert a problem of fault detection and control of a linearized constrained mechanical systems to a problem of fault detection and control of a linear time-invariant dynamic system with partially unknown inputs. The methodology appears to be mathematically elegant yet simple. The procedure or algorithm is quite straightforward and fairly easy to code or implement. As long as the necessary and sufficient conditions of the existence of an unknown input observer is met and the system is stabilizable, our proposed approach can detect and identify multiple and/or simultaneous actuator faults and sensor fault(s) almost immediately.

Chapter 5

Conclusions

In this thesis an approach for the control, fault detection and identification of constrained mechanical systems is presented. The major advantages of this state estimator or observer based analytical redundancy approach and the major contributions of this thesis can be summarized as the following:

(1) It is a systematic approach for fault detection and identification of a special class of descriptor systems that is neither infinitely observable nor completely controllable.

(2) It can detect and isolate multiple and/or simultaneous actuator and sensor faults almost immediately. The promptness of detection can be adjusted through changing the eigenvalues of the closed-loop A matrix and the eigenvalues of the observer.

(3) It is capable of distinguishing momentary failures from persistent failures. This capability exists because the FDI scheme can estimate not only the magnitude but also the shape of the faults during the entire time period in which the faults last.

(4) It can detect almost all kinds of faults. This is because that the scheme assumes no *a priori* knowledge about the nature or the mode of the failures

(5) It uses only a single observer instead of a bank of estimators.

(6) It is mathematically simple yet elegant, computationally straightforward and efficient, and relatively easy for computer simulation and/or real time implementation.

(7) A technique for numerically testing the necessary and sufficient condition under which an unknown input observer exists is found and used. Note that the following condition

$$\text{rank}\begin{bmatrix} sI_n - A & D \\ C & 0 \end{bmatrix} = n + m \qquad \forall\ s \in \mathbb{C}$$

can not be possibly numerically tested because s takes an infinite number of values. My experience/hypothesis is that testing s at all the eigenvalues of A and zero is sufficient.

(8) A modified unknown input observer whose equations are different from those contained

in a previous research work is derived.

(9) Simulations are performed using the model of a practical system - a UMS-2 robot.

(10)A modified coordinate transformation technique using a nonsingular but not orthogonal transformation matrix is developed for any mechanical system that has only one holonomic constraint. The coordinate transformation technique using an orthogonal transformation matrix which was presented in a previous research work is not applicable to the special case of a single constraint. Normal form of the linearized descriptor system model of a single constraint system (such as a UMS-2 robot) is derived in this thesis and can be shown to be different from normal forms of systems with multiple constraints.

The limitations of the proposed approach and the aspects of the topic that could be further researched by somebody else in the future can be summarized as follows:

(1) The maximum number of actuator failures and sensor failures that can be detected and identified by the approach is limited to the number of measurable outputs.

(2) Not all actuator failures can be detected and identified if they all fail at the same time.

(3) The approach requires that the considered system behaves almost linearly within an operating range, i.e., linearization of the system can be justified.

(4) A mathematical proof is not available for the experimentally correct numerical testing technique (hypothesis) with respect to the necessary and sufficient condition for the existence of an unknown input observer

On one hand, our proposed approach does not need infinite observability or complete controllability. On the other hand, for an unknown input observer to exist, at least one fairly strong (necessary and sufficient) condition has to be met. The capability of our observer based analytical redundancy approach primarily depends on the number of available outputs. The larger the number of independent outputs, the more faults we can potentially detect and identify. In the situations where a stable unknown input observer with pole placement capability does exist, our proposed approach can be very simple yet powerful.

Appendix A

Proof of Regularity of Constrained Linear Mechanical Systems

The proof uses Luenberger's shuffle algorithm (Luenberger 1978). A presentation of this algorithm and a numerical example can be found in the book of Dai (1989). Basically the algorithm involves a series of shuffling and row operations of the matrix combination $[E \quad A]$. Shuffling means the interchanging of a row of the left half of the combined matrix with that of the right half of the combined matrix. A row operation involves multiplying one row of the combined matrix with another matrix and add/subtract the product to/from another row. If the left half of the combined matrix can be made nonsingular by performing a series of alternating shuffling and row operations, then the system is said to be regular by Luenberger. Here, we present only the proof for the case of holonomic constraints because the simulation system used in chapter 4 has only one holonomic constraint. Our proof here is similar to the proof of the more general case of combined holonomic and nonholonomic constraints, which can be found in Schimidt and Muller (1991).

The linear mechanical descriptor system described by equations (2.2.23) reduces to the following form in the case of holonomic constraints:

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 \\ -K & -D & F^T \\ F & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \end{bmatrix} + \begin{bmatrix} 0 \\ f \\ 0 \end{bmatrix}$$

where

$$F = L = \frac{\partial \theta(q)}{\partial q}\bigg|_{(q_0, \dot{q}_0)} = J = \overline{H}\ , \quad G = \overline{G} = 0$$

Then

$$[E \quad A] = \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ 0 & 0 & 0 & F & 0 & 0 \end{bmatrix}$$

$$\text{shuffle 1} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ F & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Multiplying row 1 by F and subtracting the product from row 3 results in

$$\text{row operation 1} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ 0 & 0 & 0 & 0 & -F & 0 \end{bmatrix}$$

$$\text{shuffle 2} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ 0 & -F & 0 & 0 & 0 & 0 \end{bmatrix}$$

Multiplying row 2 by FM^{-1} and add the product to row 3 results in

$$\text{row operation 2} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ 0 & 0 & 0 & -FM^{-1}K & -FM^{-1}D & FM^{-1}F^T \end{bmatrix}$$

$$\text{shuffle 3} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ -FM^{-1}D & -FM^{-1}K & FM^{-1}F^T & 0 & 0 & 0 \end{bmatrix}$$

Adding $FM^{-1}D$ * row 1 and FM^{-1} * row 2 * $M^{-1}K$ to row 3 results in

$$\text{row operation 3} \Rightarrow \begin{bmatrix} I_n & 0 & 0 & 0 & I_n & 0 \\ 0 & M & 0 & -K & -D & F^T \\ 0 & 0 & FM^{-1}F^T & \Delta & \Delta & \Delta \end{bmatrix}$$

$$= \begin{bmatrix} E_3 & A_3 \end{bmatrix}$$

Since the mass matrix M is positive definite, so is M^{-1}. Then given any non-zero arbitrary vector x and its transpose x^T we have

$$x^T(FM^{-1}F^T)x = (F^T x)^T M^{-1}(F^T x) > 0$$

Therefore $FM^{-1}F^T$ is positive definite by definition and E_3 is nonsingular. The system is hence regular by Luenberger's theorem

Appendix B

Proof of Infinite Unobservability of the Augmented System

The augmented system described in chapter 2

$$\begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_e \end{bmatrix} \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dot{\lambda} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I_n & 0 & 0 \\ -K & -D & J^T & 0 \\ \bar{S} & \bar{N} & 0 & 0 \\ 0 & 0 & 0 & A_e \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \\ \lambda \\ e \end{bmatrix} + \begin{bmatrix} 0 \\ f+d \\ 0 \\ u \end{bmatrix}$$

$$y = [\, C_z \quad C_v \quad 0 \quad I_e \,] \begin{bmatrix} z \\ \dot{z} \\ \lambda \\ e \end{bmatrix}$$

is in the descriptor form

$$E\dot{x} = Ax + Bu$$
$$y = Cx$$

It is infinitely observable if and only if

$$\text{rank} \begin{bmatrix} E & A \\ 0 & E \\ 0 & C \end{bmatrix} = \text{number of rows or columns of } A + \text{rank}(E) \qquad \text{(b.1)}$$

In order to evaluate the left hand side(L.H.S.) of equation (1), we present the following theorem.

Theorem 1 The rank of a matrix will not change after the pre- or post- multiplication of a non-singular matrix

Proof: Let M be an arbitrary nonsingular matrix of order m by m

N be an arbitrary matrix of order m by p and is of rank n

P be an arbitrary nonsingular matrix of order p by p

and assuming $n \le m \le p$ without loss of generality

then using a theorem in matrix theory, we have

$$\text{rank}(M)+\text{rank}(N)-m \le \text{rank}(MN) \le \min.\{\text{rank}(M), \text{rank}(N)\}$$

or $$m+n-m \le \text{rank}(MN) \le \min.\{m,n\}$$

thus $$\text{rank}(MN) = n = \text{rank}(N)$$

This proves that the rank of a matrix is not changed by the premultiplication of a nonsingular matrix. Applying the same theorem in a similar manner will prove that

$$\text{rank}(NP)=n=\text{rank}(N)$$

and

$$\text{rank}(MNP)=\text{rank}\{(MN)P\}=\text{rank}(MN) = \text{rank}(N)$$

Therefore the theorem is valid.

Since the mess matrix M in the thesis is positive definite(see Hou et. al., 1993, second line from the top right corner on page 612), it has an inverse M^{-1}. Post-multiplying the matrix $\begin{bmatrix} E & A \\ 0 & E \\ 0 & C \end{bmatrix}$ by a nonsingular diagonal matrix containing M^{-1} in the following form preserves the rank

$$
\text{L.H.S.} = \text{rank}\begin{bmatrix} E & A \\ 0 & E \\ 0 & C \end{bmatrix} = \text{rank}\begin{bmatrix}
I_n & 0 & 0 & 0 & 0 & I_n & 0 & 0 \\
0 & M_n & 0 & 0 & -K & -D & J^T & 0 \\
0 & 0 & 0_q & 0 & \overline{S} & \overline{N} & 0 & 0 \\
0 & 0 & 0 & I_e & 0 & 0 & 0 & A_e \\
0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & M_n & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0_q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & I_e \\
0 & 0 & 0 & 0 & C_p & C_v & 0 & I_e
\end{bmatrix}
$$

$$
= \text{rank}\begin{bmatrix}
I_n & 0 & 0 & 0 & 0 & I_n & 0 & 0 \\
0 & M_n & 0 & 0 & -K & -D & J^T & 0 \\
0 & 0 & 0_q & 0 & \overline{S} & \overline{N} & 0 & 0 \\
0 & 0 & 0 & I_e & 0 & 0 & 0 & A_e \\
0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & M_n & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0_q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & I_e \\
0 & 0 & 0 & 0 & C_p & C_v & 0 & I_e
\end{bmatrix}
\begin{bmatrix}
I_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & M_n^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_q & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & I_e & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & M_n^{-1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & I_q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & I_e
\end{bmatrix}
$$

$$
= \text{rank}\begin{bmatrix}
I_n & 0 & 0 & 0 & 0 & M_n^{-1} & 0 & 0 \\
0 & I_n & 0 & 0 & -K & -DM_n^{-1} & J^T & 0 \\
0 & 0 & 0_q & 0 & \overline{S} & \overline{N}M_n^{-1} & 0 & 0 \\
0 & 0 & 0 & I_e & 0 & 0 & 0 & A_e \\
0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & I_n & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0_q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & I_e \\
0 & 0 & 0 & 0 & C_p & C_vM_n^{-1} & 0 & I_e
\end{bmatrix}
$$

Performing elementary row operations on the above matrix yields

$$\text{L.H.S.} = \text{rank}\begin{bmatrix} I_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0_q & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_e & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0_q & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_e \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= (n + n + e) + (n + n + e)$$

$$= 2(2n + e)$$

The right hand side(R.H.S.) of equation (1) is evaluated as

$$\text{R.H.S.} = (n + n + q + e) + \text{rank}\begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & M_n & 0 & 0 \\ 0 & 0 & 0_q & 0 \\ 0 & 0 & 0 & I_e \end{bmatrix}$$

$$= (2n + q + e) + (n + n + e)$$

$$= 2(2n + e) + q$$

Obviously L.H.S. of equation b.1 $\neq$ R.H.S. of equation b.1, the necessary and sufficient condition for infinite observability as expressed by equation b.1 does not hold. Therefore the system is infinitely unobservable.

Appendix C

Simulation Program Source Code

```
%***********************************************************
%                    APPENDIX C
%              SIMULATION PROGRAM SOURCE CODE
%           Simulation of a UMS-2 Robot in MATLAB
%***********************************************************

% this simulation detects and identifies 2 actuator failures
% and 1 sensor failure

Na = 2 % Na: number of actuator failures
Ne = 1 % Ne: number of sensor failure


% specify matrices used in linearized descriptor system model

M = [0.9 0 0
     0 3 0
     0 0 2 ]

D = zeros(3)

K = [0.3836 0 3.7207
       0    0  0
    -0.2793 0 -2]

L = [0.2418 1 1.3289]

In = eye(3)

J = [0.2418 1 1.3289] % Jacobian J = L in the holonomic case

% nonsingular coordinate transformation begins

T = [0 4.1356 0
     1  -1    1
 -0.7525 0    0]    % transformation matrix
```

```
LT=L*T                              % verify that LT = [0 0 1]

TtMT = T'*M*inv(T')

TtDT = T'*D*inv(T')

TtKT = T'*K*inv(T')

q = 1

n=3

M11=TtMT(1:n-q,1:n-q)

K11=TtKT(1:n-q,1:n-q)

D11=TtDT(1:n-q,1:n-q)

M12 = TtMT(1:n-q,(n-q+1):n)

K12 = TtKT(1:n-q,(n-q+1):n)

D12 = TtDT(1:n-q,(n-q+1):n)

LTtInv = L*inv(T');

L11 = LTtInv(:,1:n-q)

L12 = LTtInv(:,n-q+1:n)

A21 = -inv(M11-M12*L11/L12)*(K11-K12*L11/L12)

A22 = -inv(M11-M12*L11/L12)*(D11-D12*L11/L12)

A=[zeros(n-q) eye(n-q)
      A21    A22   ]

B=[zeros(n-q); inv(M11-M12*L11/L12)]

% nonsingular coordinate transformation ends


% perform controllability test on the normal form representation
```

```
ContrMatrix=ctrb(A,B)

ContrMatrixRank=rank(ContrMatrix)


%specify measurement matrix used in original system representation

Cm = [1 0 0 0 0 0
      0 1 0 0 0 0
      0 0 1 0 1 1
      0 0 0 1 0 0]

Cp = Cm(:,1:n)

Cv = Cm(:,(n+1):(2*n))


% obtain output matrices used in normal form representation

C1 = Cp*inv(T')*[eye(n-q);L11/L12]

C3 = Cv*inv(T')*[eye(n-q);L11/L12]

C=[C1 C3]


% specify actuator fault distribution matrix

D = B


% perform observability test

ObservMatrix = obsv(A,[C1 C3])

ObservMatrixRank = rank(ObservMatrix)


% test observer existence conditions

C
D
CD = C*D

Rank_C = rank(C)
```

```
Rank_D =rank(D)

Rank_CD = rank(C*D)

Rank_D-Rank_CD

% Redefine the above matrices as open-loop matrices
% use subscript o to denote open-loop

Ao=A
Bo=B
Co=C
Do=D

CoDo = Co*Do

% Stabilize the open-loop system using state feedback technique

Pc = [-6+2i;-6-2i;-5+4i;-5-4i]   % choose closed-loop poles

Kc = place(Ao,Bo,Pc)% state feedback gain matrix

Ac = Ao-Bo*Kc

eigenvalue = zeros(n,1);

Eig_Val_Ac = eig(Ac)

% Augment the system to accomodate sensor failure

% define open-loop system matrices Aao, Bao, Cao, Dao, Kao

Aao = [Ao zeros(4,Ne);zeros(Ne,4) -5]
        % set additional eigenvalue at -5

Bao = [Bo; zeros(Ne,(n-q))]

% specify sensor failure distribution vector

Ef = [1,0,0,0]
```

```
Cao = [Co Ef]

Dao = [Do zeros(4,1); zeros(1,2) eye(1)]

CaoDao = Cao*Dao


% check observability of augmented open-loop system

Observability_Augm = obsv(Aao,Cao)

Observ_Augm_Rank = rank(Observability_Augm)


% check rank conditions

Rank_Dao = rank(Dao)

Rank_CaoDao = rank(Cao*Dao)


% Obtain augmented closed-loop dynamic model

Karb = ones(2,1)
% feedback gain matrix used to stabilize augmented system
% It can be proved that this matrix can be chosen arbitrarily

Kao = [Kc Karb]

Aac = Aao - Bao*Kao

Eig_Val_Aac = zeros(5,1);

Eig_Val_Aac = eig(Aac)

%Redefine system order using closed-loop augmented representation

n = 5   % number of state variables

p = 4   % number of outputs

m = 3   % number of combined actuator failures and sensor failure


% Check augmented closed-loop system observer existence condition
```

```
Ranks_wrt_Eig_Val = zeros(n,1);

Observability_Test = [-Aac,Dao;Cao,zeros(p,m)]

for i = 1:n

Ranks_wrt_Eig_Val(i) = rank([Eig_Val_Aac(i)*eye(n)-Aac,Dao;Cao,zeros(p,m)]);

end

Rank_wrt_Zero_Eig_Val = rank([-Aac,Dao;Cao,zeros(p,m)])

Ranks_wrt_Eig_Val


% check rank conditions

Rank_Dao = rank(Dao)

Rank_CaoDao = rank(Cao*Dao)


% check observability of augmented closed-loop system

% Observability_Augm = obsv(Aac,Cao)

% Observ_Augm_Rank = rank(Observability_Augm)

% [-Aac,Dao;Cao,zeros(p,m)]

% rank([-Aac,Dao;Cao,zeros(p,m)])


% proceed to obtaining reduced order observer

Q=[zeros(n-p,p),eye(n-p)]

P=[Q;Cao]     % transformation matrix to bring C to [O,I] form

Pinv=inv(P)

% First transformation is now taking place

As=P*Aac*Pinv

Bs=P*Bao
```

```
Cs=Cao*Pinv

Ds=P*Dao

% Convert Cs=[0 Ip] to Cn=[0 Cp] to deal with the invertibility of Ds3

% The new transformation matrix Pn is chosen such that Dn3 is
% invertible. This is accomplished by switching row 2 and
% row 3 in Ds

Pn = [1 0 0 0 0
      0 0 1 0 0
      0 1 0 0 0
      0 0 0 1 0
      0 0 0 0 1]

% the new representation are defined by An, Bn, Cn, & Dn.
% n denotes new

An = Pn*As*inv(Pn)

Bn = Pn*Bs

Cn = Cs*inv(Pn)

Dn = Pn*Ds

% Obtain the Cp in Cn=[0 Cp] and Cp1 & Cp2 in x2=Cp1*y & x3=Cp2*y

Cp = Cn(:,(n-p+1):n)

Cpinv = inv(Cp)

Cp1 = Cpinv(1:(p-m),:)

Cp2 = Cpinv((p-m+1):p,:)


A1=An(1:n-p,:)

A2=An(n-p+1:n-m,:)

A3=An(n-m+1:n,:)

B1=Bn(1:n-p,:)
```

```
B2=Bn(n-p+1:n-m,:)

B3=Bn(n-m+1:n,:)

D1=Dn(1:n-p,:)

D2=Dn(n-p+1:n-m,:)

D3=Dn(n-m+1:n,:)

D3inv=inv(D3)

A1bar=A1-D1*D3inv*A3

A2bar=A2-D2*D3inv*A3

B1bar=B1-D1*D3inv*B3

B2bar=B2-D2*D3inv*B3

A11bar=A1bar(:,1:n-p)

A12bar=A1bar(:,n-p+1:n-m)

A13bar=A1bar(:,n-m+1:n)

A21bar=A2bar(:,1:n-p)

A22bar=A2bar(:,n-p+1:n-m)

A23bar=A2bar(:,n-m+1:n)

% obtain observer in the form of (3.5.1.19)-(3.5.1.24) of thesis

Pole_observer = -6     % choose observer pole at -6

M = place(A11bar,A21bar,Pole_observer)     % observer gain matrix

F = A11bar-M*A21bar

E1 = (A11bar-M*A21bar)*(D1*D3inv*Cp2-M*(Cp1-D2*D3inv*Cp2));

E2 = ((A12bar*Cp1+A13bar*Cp2)-M*(A22bar*Cp1+A23bar*Cp2));

E = E1+E2 % E is too long to be typed in one row
```

```
%continuing expression in 2nd row would have resulted in E=E1

L = B1bar-M*B2bar

N = D1*D3inv*Cp2-M*(Cp1-D2*D3inv*Cp2)


% PREPARE FOR LINEAR DYNAMIC SYSTEM SIMULATION


% DEFINE SAMPLING PERIOD

Ts = [0:0.1:19.9]';     % sample taken at 0.1 sec. interval
                        % for 20 seconds


% specify applied generalized forces or known inputs

u1 = 7*ones(200,1);

u2 = 8*ones(200,1);

u3 = 9*ones(200,1);

Tt = T'            % transpose of nonsingular transformation matrix

U = (Tt(1:2,:)*[u1';u2';u3'])';          % known input vector

% STARTING SIMULATION #1

% specify arbitrary actuator failures and sensor failure
% for the sake of simulation

d1 = [zeros(30,1); 0.5*ones(170,1)]; % actuator #1 failure

d2 = [zeros(30,1); 0.4*ones(170,1)]; % actuator #2 failure

d3 = [zeros(30,1); 0.0*ones(170,1)]; % actuator #3 faultless

Tsub = [0.0     1.0
        4.1356  -1.0]

d = (Tsub*[d1';d2'])'; % unknown input matrix

f1 = [zeros(30,1); 0.5*ones(170,1)]; % sensor #1 failure
```

```
f = f1;

% start simulation of continuous time state-space model of
% the unaugmented closed-loop system

Xc0 =[0;0;0;0]% arbitrary initial condition of the state vector

[Yc, Xc] = lsim(Ac,[Bo,Do],Co,zeros(p,Na+Na),[U,d],Ts,Xc0);

% Get output vector for the augmented closed-loop system

Yac = Yc + (Ef*f)';

% Start simulation of reduced order observer

W0 = 0          % arbitrary initial condition of reduced order observer

W = zeros(1,200);

[Yobserver,W]=lsim(F,[E,L],0,zeros(1,6),[Yac, U],Ts,W0);

% Estimate state vector of twice transformed representation Xn

Yn = Yac;       % output doesn't change during transformation

Xn = ([1;0;0;0;0]*W' + [N;Cp1;Cp2]*Yn')';

% estimate state vector of once transformed representation Xs

Xs = (inv(Pn)*(Xn)')';

% estimate state vector of augmented closed-loop system Xac

Xac = (inv(P)*(Xs)')';

% estimate state vector of un-augmented closed-loop system Xc

Xc = Xac(:,1:n-1);

% estimate sensor fault failure
```

```
f1_estimate = Xac(:,n);

% plotting sensor fault and its estimate against time

figure(1)

plot(Ts,f1,'g-',Ts,f1_estimate,'r--',Ts,f1,'g-',Ts,f1_estimate,'y.')

title('Figure 1 - simulation #1: sensor 1 failure and its estimate')

xlabel('time(s)')

print figure1 -dps

% obtain actuator failure estimates using least-square approach

S = zeros(n-1,199);

v = zeros(Na,199);          % specify the dimension for
                            % unknown inputs estimates

for k = 1:199

S1 = Ac*inv(expm(Ac*0.1)-eye(n-1));

S(:,k)=S1*(Xc(k+1,:)'-expm(Ac*0.1)*Xc(k,:)');

v(:,k) = inv(Do'*Do)*Do'*(S(:,k)-Bo*U(k,:)');

end

d1_bar_estimate = v(1,:)';

d2_bar_estimate = v(2,:)';

Tt_sub = Tt(1:2,1:2)              % upper left sub matrix of Tt

d_estimate = inv(Tsub)*[d1_bar_estimate';d2_bar_estimate'];

d1_estimate = d_estimate(1,:)';          % estimate of actuator 1 fault

d2_estimate = d_estimate(2,:)';          % estimate of actuator 2 fault
```

```
d1_differential = d1(1:199) - d1_estimate;

d2_differential = d2(1:199) - d2_estimate;

% Plotting actuator failures and their estimates against time

T1 = [0:0.1:19.8]';

d1t = d1(1:199,1);

d2t = d2(1:199,1);

figure(2)

plot(T1,d1t,'g-',T1,d1_estimate,'r--',T1,d1t,'g-',T1,d1_estimate,'y.')

title('Figure 2 - simulation #1: actuator 1 failure and its estimate')

xlabel('time(s)')

print figure2 -dps

figure(3)

plot(T1,d2t,'g-',T1,d2_estimate,'r--',T1,d2t,'g-',T1,d2_estimate,'y.')

title('Figure 3 - simulation #1: actuator 2 failure and its estimate')

xlabel('time(s)')

print figure3 -dps


% STARTING SIMULATION #2

% specify arbitrary actuator failures and sensor failure
% for the sake of simulation

d1 = [zeros(30,1); 0.5*ones(170,1)]; % actuator #1 failure

d2 = [zeros(30,1); 0.0*ones(170,1)]; % actuator #2 faultless

d3 = [zeros(30,1); 0.4*ones(170,1)]; % actuator #3 failure

Tsub = [0    -0.7525
        4.1356   0]
```

```
d = (Tsub*[d1';d3'])'; % unknown input matrix

f1 = [zeros(30,1); 0.5*ones(170,1)]; % sensor #1 failure

f = f1;

% start simulation of continuous time state-space model of
% the unaugmented closed-loop system

Xc0 =[0;0;0;0]% arbitrary initial condition of the state vector

[Yc, Xc] = lsim(Ac,[Bo,Do],Co,zeros(p,Na+Na),[U,d],Ts;Xc0);

% Get output vector for the augmented closed-loop system

Yac = Yc + (Ef*f)';

% Start simulation of reduced order observer

W0 = 0            % arbitrary initial condition of reduced order observer

W = zeros(1,200);

[Yobserver,W]=lsim(F,[E,L],0,zeros(1,6),[Yac, U],Ts,W0);

% Estimate state vector of twice transformed representation Xn

Yn = Yac;    % output doesn't change during transformation

Xn = ([1,0,0,0,0]*W' + [N;Cp1;Cp2]*Yn')';

% estimate state vector of once transformed representation Xs

Xs = (inv(Pn)*(Xn)')';

% estimate state vector of augmented closed-loop system Xac

Xac = (inv(P)*(Xs)')';

% estimate state vector of un-augmented closed-loop system Xc
```

```
Xc = Xac(:,1:n-1);

% estimate sensor fault failure
f1_estimate = Xac(:,n);

% plotting sensor fault and its estimate against time
figure(4)
plot(Ts,f1,'g-',Ts,f1_estimate,'r--',Ts,f1,'g-',Ts,f1_estimate,'y.')
title('Figure 4 - simulation #2: sensor 1 failure and its estimate')
xlabel('time(s)')
print figure4 -dps

% obtain actuator failure estimates using least-square approach
S = zeros(n-1,199);
v = zeros(Na,199);        % specify the dimension for
                          % unknown inputs estimates
for k = 1:199
S1 = Ac*inv(expm(Ac*0.1)-eye(n-1));
S(:,k)=S1*(Xc(k+1,:)'-expm(Ac*0.1)*Xc(k,:)');
v(:,k) = inv(Do'*Do)*Do'*(S(:,k)-Bo*U(k,:)');
end
d1_bar_estimate = v(1,:)';
d2_bar_estimate = v(2,:)';
Tt_sub = Tt(1:2,1:2)              % upper left sub matrix of Tt
d_estimate = inv(Tsub)*[d1_bar_estimate',d2_bar_estimate'];
```

```
d1_estimate = d_estimate(1,:)';          % estimate of actuator 1 fault

d3_estimate = d_estimate(2,:)';          % estimate of actuator 3 fault

d1_differential = d1(1:199) - d1_estimate;

d3_differential = d3(1:199) - d3_estimate;

% Plotting actuator failures and their estimates against time

T1 = [0:0.1:19.8]';

d1t = d1(1:199,1);

d3t = d3(1:199,1);

figure(5)

plot(T1,d1t,'g-',T1,d1_estimate,'r--',T1,d1t,'g-',T1,d1_estimate,'y.')

title('Figure 5 - simulation #2: actuator 1 failure and its estimate')

xlabel('time(s)')

print figure5 -dps

figure(6)

plot(T1,d3t,'g-',T1,d3_estimate,'r--',T1,d3t,'g-',T1,d3_estimate,'y.')

title('Figure 6 - simulation #2: actuator 3 failure and its estimate')

xlabel('time(s)')

print figure6 -dps

% END OF MATLAB SOURCE CODE
```

Figure 1 - simulation #1: sensor 1 failure and its estimate

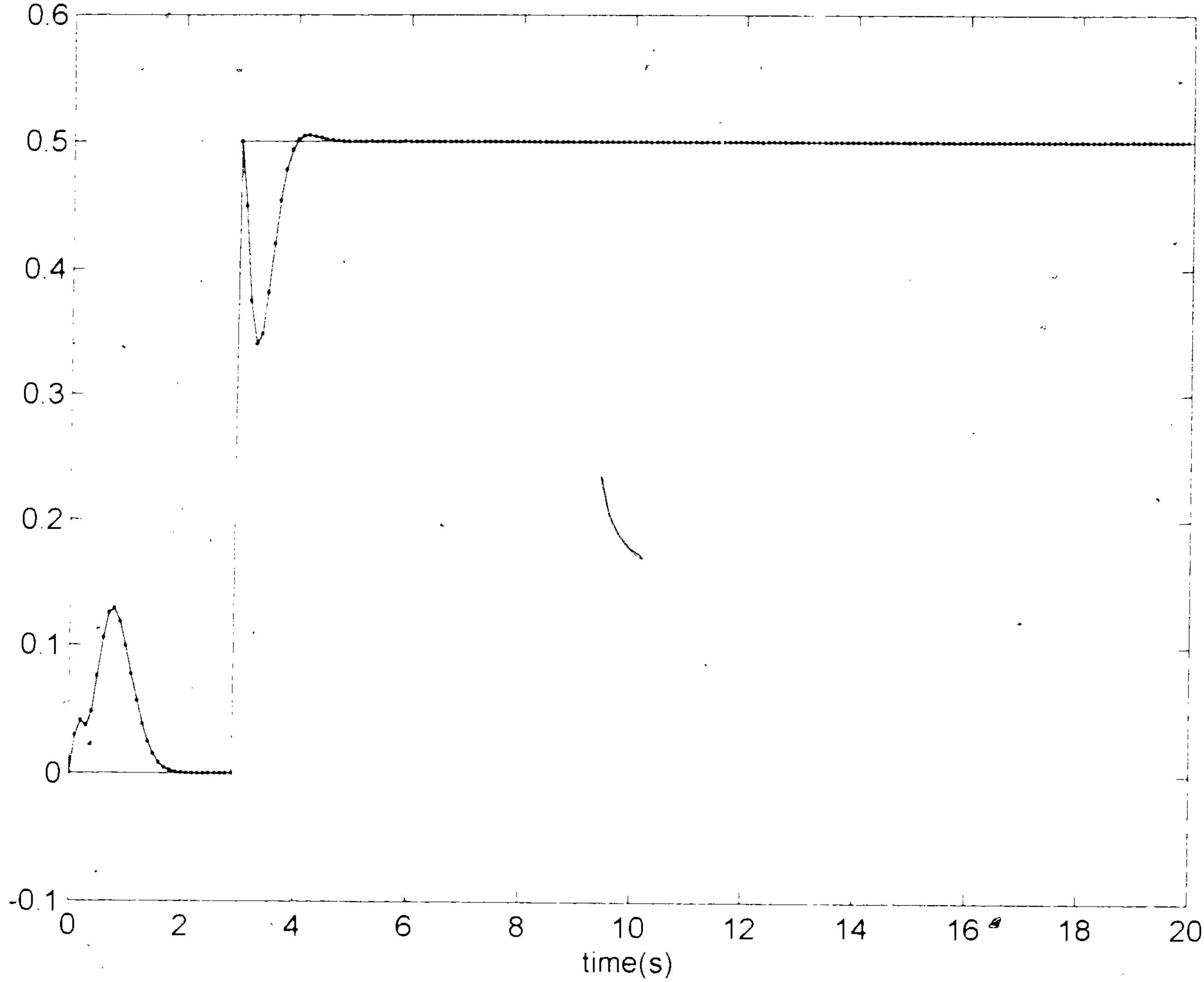


Figure 2 - simulation #1: actuator 1 failure and its estimate

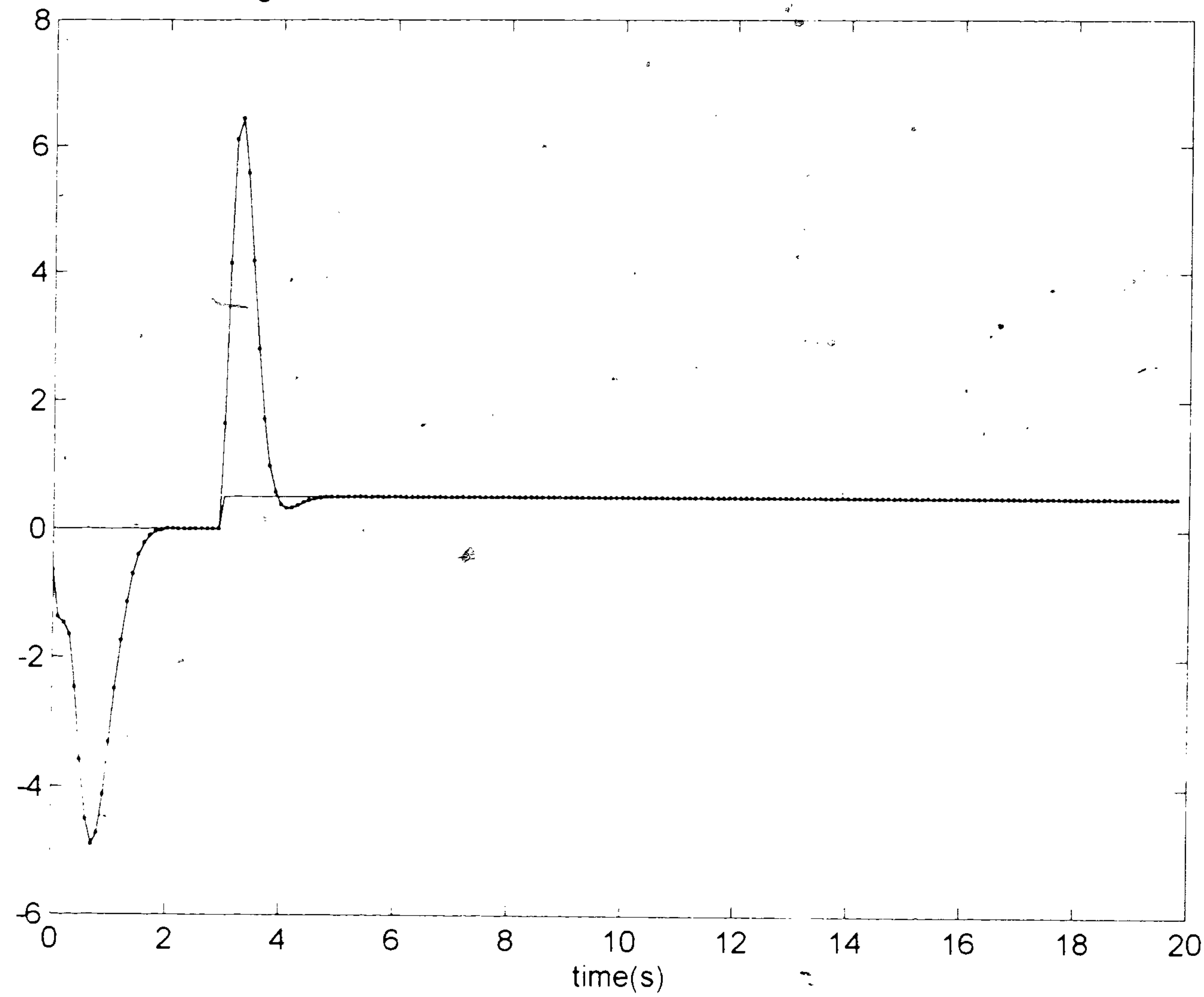


Figure 3 - simulation #1: actuator 2 failure and its estimate

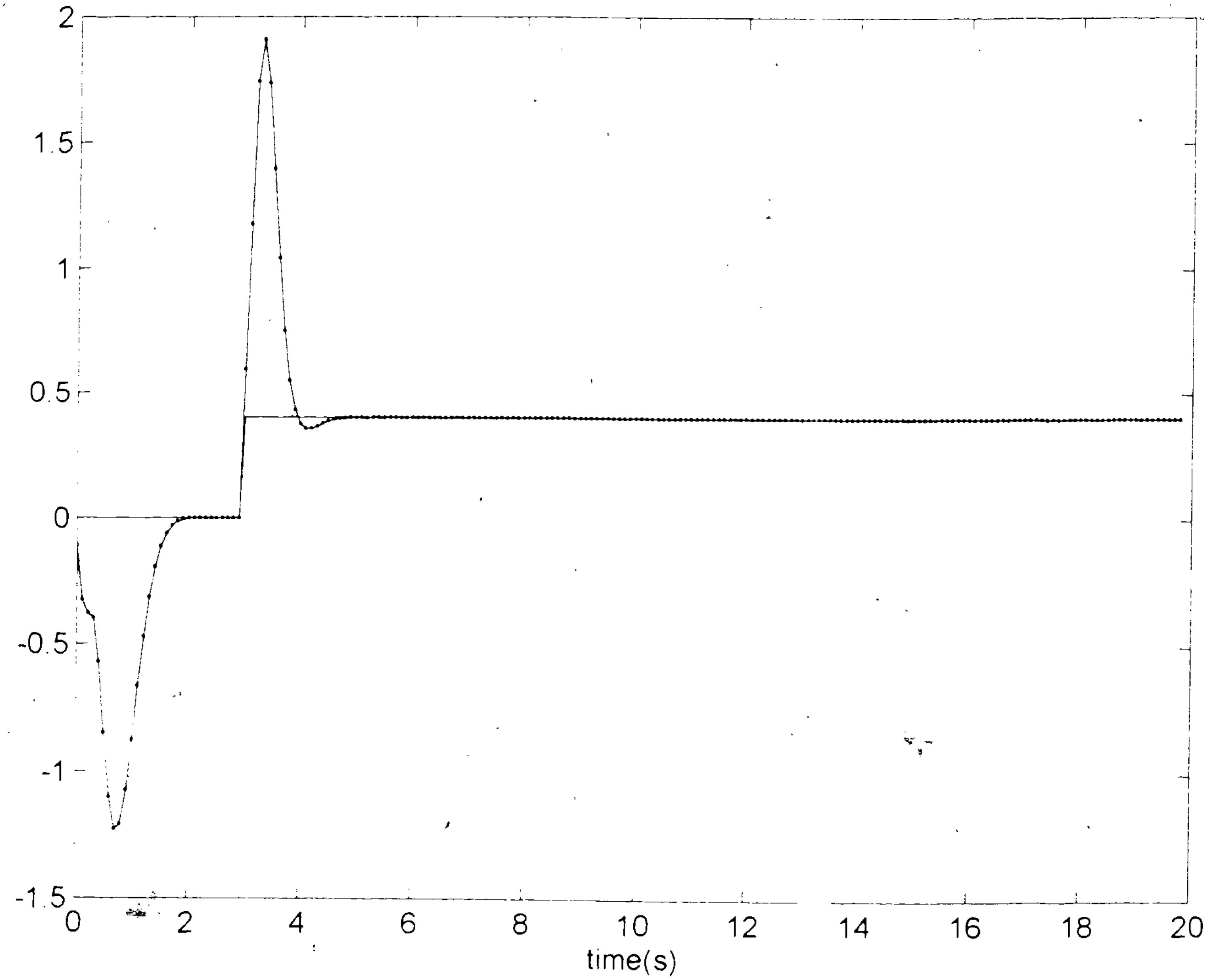
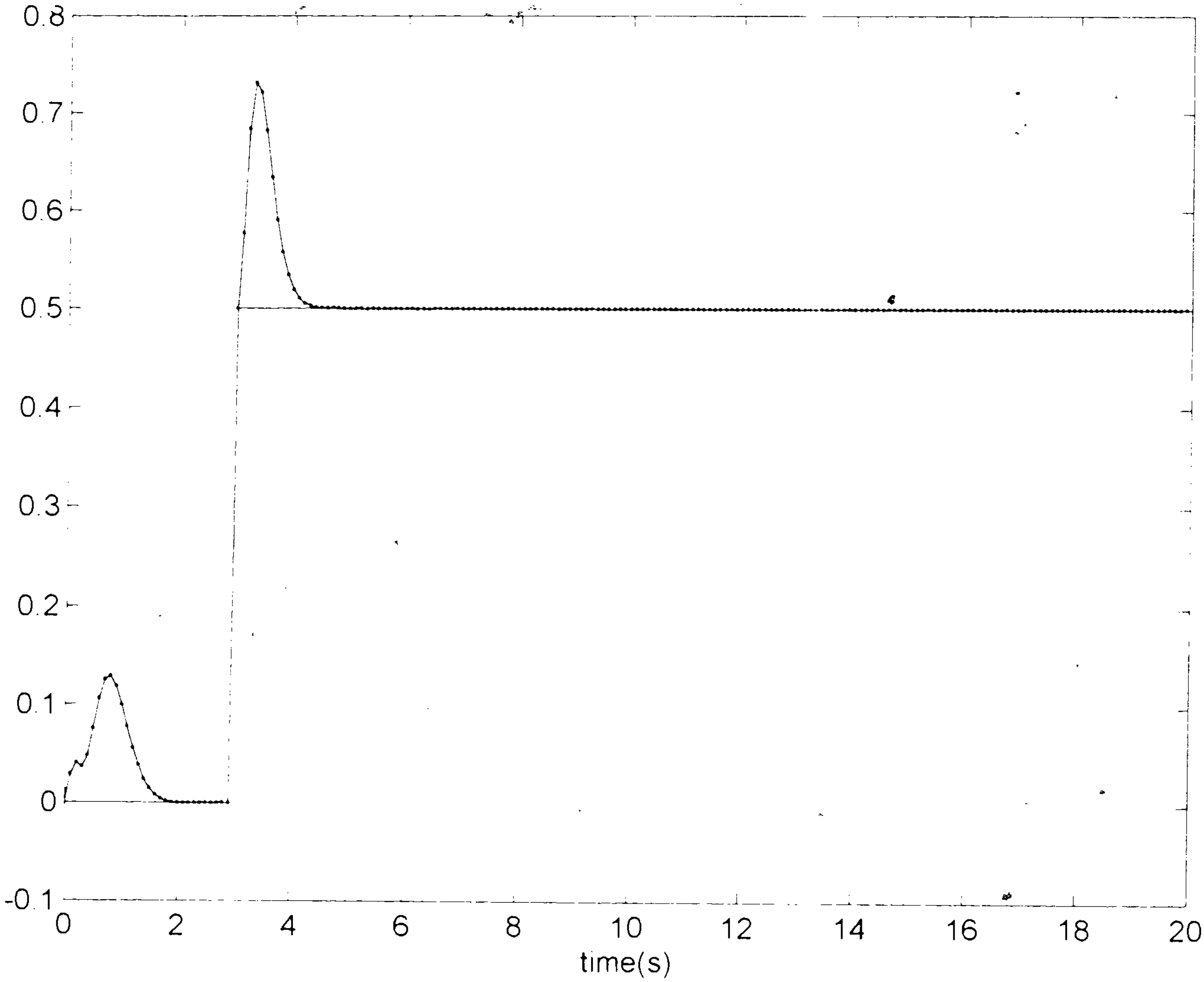


Figure 4 - simulation #2: sensor 1 failure and its estimate



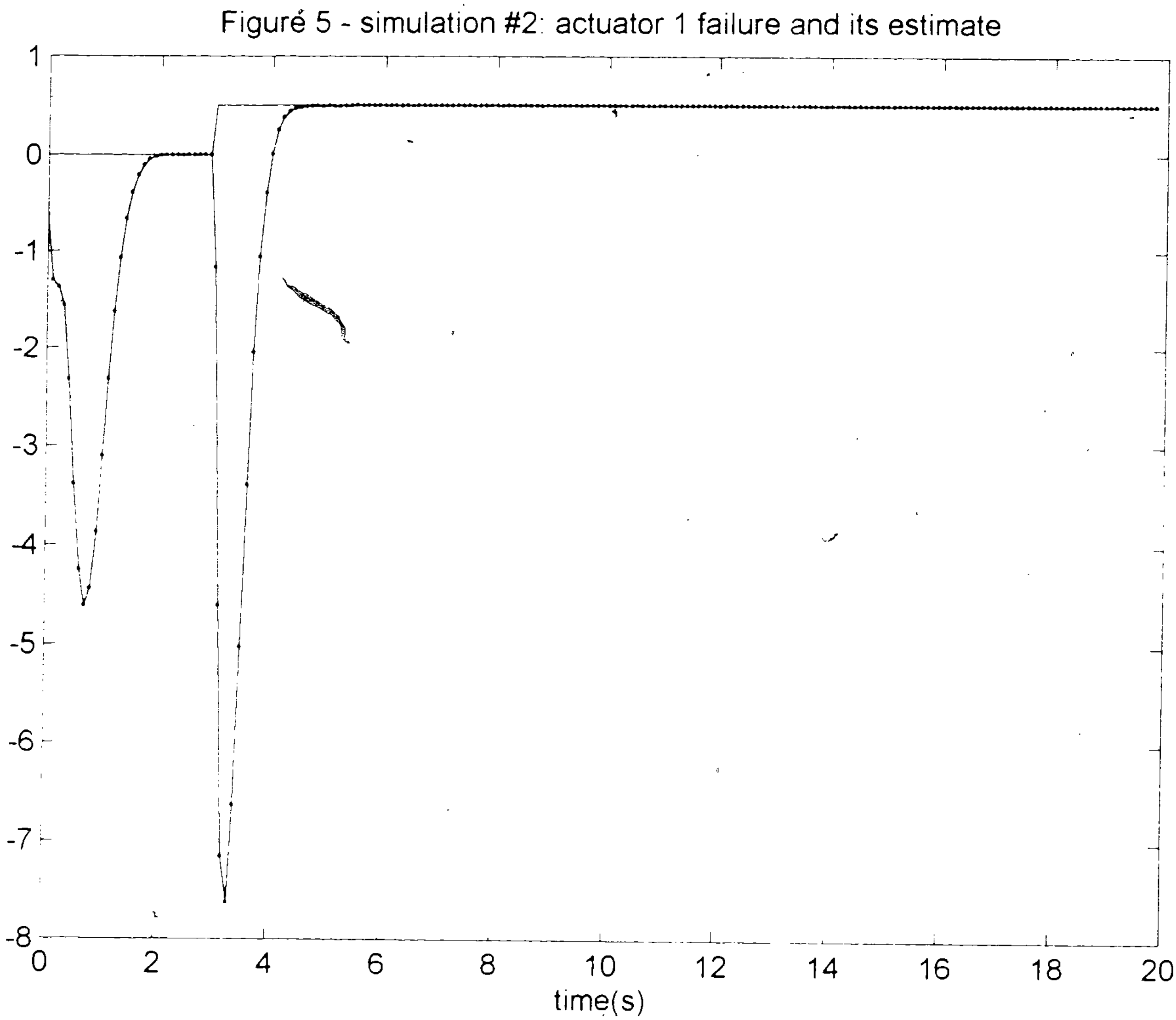
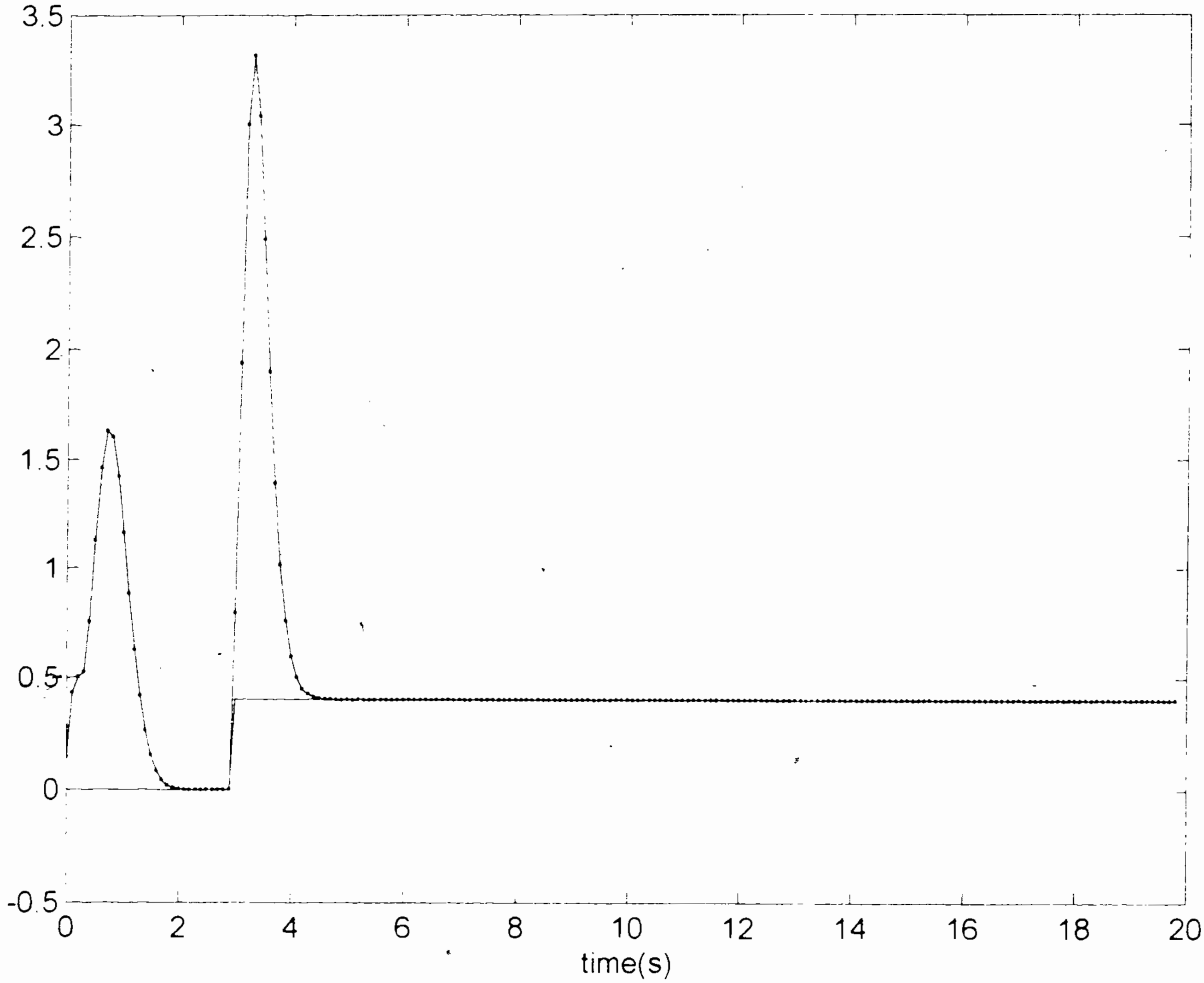


Figure 5 - simulation #2: actuator 1 failure and its estimate
1
0
-1
-2
-3
-4
-5
-6
-7
-8
0
2
4
6
8
10
12
14
16
18
20
time(s)

Figure 6 - simulation #2: actuator 3 failure and its estimate



References

References are listed in alphabetical order by last names

Chen, C.T., 1984
Linear System Theory and Design, **New York: HRW Publishing**

Clark, R.N., 1978
"Instrument Fault Detection", **IEEE Transactions on Aerospace and Electronic Systems,** AES-14, 456-465

Clark, R.N., 1979
"The Dedicated Observer Approach to Instrument Failure Detection", **Porch. 15-th IEEE CDC Conference**, Fort Lauderdale, Fla., pp. 237-241, Dec. 1979

Darouach, M., 1994
"On the Novel Approach to the Design of Unknown Input Observers", **IEEE Transactions on Automatic Control**, Vol. 39, No. 3, pp. 698-699

Dai, L., 1989
Singular Control Systems, Lecture Notes in Control and Information Sciences, **Springer, Berlin**

Emami-Naeini, A., Akhter, M.M., and Rock, S.M., 1988
"Effect of Model Uncertainties in Failure Detection: The Threshold Method", **IEEE Transactions on Automatic Control**, 33, pp. 1106-1115

Frank, P.M., 1990,
"Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy-A Survey and Some New Results", **Automatica**, Vol. 26, pp. 459-474

Frank, P.M., and Keller, L., 1980
"Sensitivity Discriminating Observer Design for Instrument Failure Detection", **IEEE Transactions on Aerospace and Electronic Systems**, AES-16, 460-467.

Goldstein, H., 1980
Classical Mechanics, **Addison-Wesley**

Greenwood, D.T., 1965
Principles of Dynamics, **Prentice Hall, Englewood Cliffs**

Guan, Yuping, and Saif, Mehrdad, 1991
"A Novel Approach to the Design of Unknown Input Observers", **IEEE Transactions on Automatic Control**, Vol. 36, No. 5, pp. 632-635

Hou, M., and Muller, P.C. 1992
"Design of Observer for Linear Systems with Unknown Inputs", **IEEE Transactions on Automatic Control**, Vol. AC-37, pp. 871-875

Hou, M., Schmidt, Th., Schupphaus, R., and Muller, P.C. 1993
"Normal Form and Luenberger Observer for Linear Mechanical Descriptor Systems", **Journal of Dynamic Systems, Measurement, and Control**, Vol. 110, pp. 255-165

Isermann, R., 1984
"Process Fault Detection Based on Modeling and Estimation Methods - A Survey", **Automatica**, Vol. 20, No. 4, pp. 387-404

Johnson, C.D., 1975
"On Observers with Unknown and Inaccessible Inputs", **International Journal of Control,** Vol. 21, pp. 825-831

Kane, T.R., and Levinson, D.A., 1985
Dynamics: Theory and Applications, **McGraw-Hill, New York**

Kitamura, M., 1980
"Detection of Sensor Failures in Nuclear Plant Using Analytic Redundancy", **Trans. Am. Nucl. Soc.**, Vol., 34, 581-583

Kitamura, M., Matsubara, K., and Oguma, R., 1979
"Identifiability of Dynamics of a Boiling Water Reactor using Autoregressive Modeling", **Nucl. Sci. Engng.**, Vol. 70, 106-110

Kitamura, M., Washio, T., Kotajima, K. and Sugiyama, K., 1985
"Small-sample Modeling Method for Nonstationary Reactor Noise Analysis", **Annals Nucl. Ennergy,** Vol. 12, 399-407

Kudva, P., Viswanadham, H., and Ramakrishna, A., 1980
"Observers for Linear Systems with Unknown Inputs", **IEEE Transactions on Automatic Control,** Vol. AC-25, pp. 113-115

Kurek, J. E., 1983
"The State Vector Reconstruction for Linear Systems with Unknowy Inputs", **IEEE Transactions on Automatic Control,** Vol. AC-28, pp. 1120-1122

Lewis, F.L., 1986
"A Survey of Linear Singular Systems", **Circuits, Systems, and Signal Processing,** Vol. 5, pp. 3-36

Luenberg, D.G., 1974
"Dynamic Equations in Descriptor Forms", **IEEE Transactions on Automatic Control,** Vol. AC-19, pp. 201-208

Luenberg, D.G., 1978
"Time-Invariant Descriptor Systems", **Automatica, Vol. 14**, pp. 473-480

Meditch, J.S. and Hostetter, G.H, 1974
"Observers for Systems with Unknown and Inaccessible Inputs", **International Journal of Control,** Vol. 19, pp. 473-480

Meirovitch, L., 1970
Method of Analytical Dynamics, **McGraw-Hill, New York**

Merrill, W.C., 1985
"Sensor Failure Detection for Jet Engines Using Analytical Redundancy", **AIAA Journal of Guidance, Control and Dynamics,** 8, pp.673-682

Mill, James, and Goldenberg, Andrew, 1989
"Force and Position Control of Manipulators During Constrained Motion Tasks", **IEEE Transactions on Robotics and Automation,** Vol. 5, No. 1, pp. 30-46

Neimark, J.I, and Fufaev, N.A., 1972
Dynamics of Nonholonomic Systems, **American Mathematical Society**

Pattern, R., Frank, P. and Clark, R.(Editors), 1989
Fault Diagnosis in Dynamic Systems-Theory and Application, **Prentice Hall**

Pau, L.P., 1986
"Survey of Expert Systems for Fault Detection, Test Generation and Maintenance", **Expert Systems,** Vol. 3, 100-111

Rugh, W.J., 1993
Linear System Theory, **Prentice-Hall, New Jersey**

Saif, Mehrdad, and Guan, Yuping.,1993
"A New Approach to Robust Fault Detection and Identification", **IEEE Transactions on Aerospace and Electronic Systems,** Vol. 29, No. 3, pp. 685-695

Schmidt, Th., and Muller, P.C., 1991
"Parameterschatzung komplexermechanischer Regelungssysteme mit Zwangsbedingungen", **Technical Report, Safety Control Engineering, University of Wuppertal**

Shin, K.C., and Kabamba, P.T., 1988
"Observation and Estimation in Linear Descriptor Systems with Application to Constrained Dynamical Systems", **Journal of Dynamic Systems, Measurement, and Control.** Vol 115, pp. 611-620

Swisher, George M., 1976
Introduction to Linear Systems Analysis, **Matrix Publishers Inc., Chesterland, Ohio**

Tzafestas, S.G., 1987
"Artificial Intelligence Techniques in Control: An Overview", **Proc. of the IMACS Symp. on AI, Expert Systems and Languages in Modelling and Simulation, Barcelona, C. Kulikowski and G. Ferrate (eds)**

Tzafestas, S.G., 1989
Knowledge Based System Diagnosis, Supervision, and Control, **Plenum, New York-London**

Viswanadham, N., and Srichander, R., 1987
"Fault Detection Using Unknown Input Observers", **Control Theory and Advanced Technology (C-TAT),** 3, 91-101

Vukobratovic, K.M., and Potkonjak, V., 1982
Dynamics of Manipulator Robots: Theory and Application, **Springer Verlag, Berlin**

Wang, S.H., Davison, E.J., and Dorato, P., 1975
"Observing the States of Systems with Unmeasurable Disturbances", **IEEE Transactions on Automatic Control,** Vol. AC-20, pp 716-717

Willsky, A. S., 1976
"A Survey of Design Methods for Failure Detection in Dynamic Systems", **Automatica,** Vol. 12, pp. 601-611

Yang, F., Wilde, and R.W., 1988
"Observers for Systems with Unknown Inputs", **IEEE Transactions on Automatic Control,** Vol 33, NO. 7, pp. 677-681

Yip, E. L., and Sincovec, R.F., 1981
"Solvability, Controllability, and Observability of Continious Descriptor Systems", **IEEE Transactions on Automatic Control,** Vol AC-26, NO 3, pp 702-707