Analysis of a sliding observer for a beam with one-sided spring

Citation for published version (APA):
Blansjaar, J. (1997). *Analysis of a sliding observer for a beam with one-sided spring*. (DCT rapporten; Vol. 1997.065). Technische Universiteit Eindhoven.

Document status and date:
Published: 01/01/1997

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Analysis of a Sliding Observer for a Beam with One-Sided Spring

J. Blansjaar

WFW Report 97.065

Professor: Prof.dr.ir. J.J. Kok
Supervisors: Ir. M.F. Heertjes
Dr.ir. M.J.G. van de Molengraft

Eindhoven University of Technology
Faculty of Mechanical Engineering

Section: System & Control Engineering

Internship Report, August 1997

Luctor et Emergo

Contents

Abstract

Vibration reduction in harmonically excited nonlinear systems is a research area with quite some applications. Reduction of oscillations in ships colliding at quay sides or diminishing the rattling of gear are just two examples which will decrease wear and the loud noises that accompany such vibrations. Depending on the excitation frequency responses of various frequencies and amplitudes might exist for a harmonically excited system with a local nonlinearity. Naturally, only the stable response will occur in a freely oscillating system. A reduction of vibration amplitudes can be achieved when the amplitude of the unstable response is considerably smaller than of the coexisting stable response. An external force can then be applied to force the system to its unstable response of smaller amplitude. Theoretically the necessary external force will diminish as soon as the unstable response is attained since it should require little energy to keep the system at that (unstable) solution. A beam with a one-sided spring is an example of such a system with local nonlinearity. It has been achieved to control the vibration of this system. The current objective is to further decrease the control force required to bring and maintain the beam in its unstable response. Possibly a sliding observer can achieve such a decrement by improving the state estimation of the system.

A sliding mode observer (SMO) consists of a linear estimator (a Luenberger observer) complemented with some nonlinear terms to compensate for the system nonlinearities it therefore is suitable for the observation of systems with added nonlinearities. The simulation trials that were held indeed showed an improvement of the state reconstruction and the required control force was diminished by the sliding observer. However, a major obstacle for a more significant improvement is the fact that a measurement of displacement of only two DOFs is applied: the SMO needs the displacement of all DOFs to make its estimation. The best result was attained when applying a sliding observer with a Kalman filter as its linear part. Increasing the sample frequency also improved the estimation.

Symbols & Abbreviations

Symbols

A	linear system matrix
A_{nl}	nonlinear system matrix
ΔA_{nl}	difference between linear and nonlinear system matrix
β	sliding motion smoothening boundary layer
$\underline{b}$	input distribution vector of 1-DOF system
B	input distribution matrix of 3-DOF system
$\underline{c}^T$	measurement column of 1-DOF system
C	measurement matrix of 3-DOF system
η	boundary layer of the controller
$\underline{e}_{\hat{y}}$	difference between estimated and true measurement
$\underline{e}_x$	tracking error
$\underline{e}_{\hat{x}}$	estimation error
$\underline{\varphi}$	boundary layer vector of the SMO
f_s	simulation frequency
H	input distribution matrix of the 3-DOF movement equations
k	linear stiffness element of the 1-DOF system matrix
$\underline{k}_{lo}$	Luenberger observer of the 1-DOF system
k_{ls}	leaf spring stiffness
k_{nl}	nonlinear stiffness element of 1-DOF system matrix
$\underline{k}_{smo}$	sliding mode observer gain of the 1-DOF system
K	stiffness matrix of 3-DOF movement equations
K_{lo}	Luenberger observer gain matrix
K_o	optimal Kalman filter gain matrix
K_{smo}	sliding mode observer gain matrix of the 3-DOF system
λ	controller parameter
m	mass of 1-DOF system
m_e	eccentric mass of 3-DOF system excitation
n_e	number of elements in signal vector
$\underline{q}$	DOF column
r_e	eccentricity radius of eccentric mass
σ	controller parameter
S	sliding surface transformation matrix
t	time
$\underline{t}$	time vector
t_s	simulation start time
t_f	final simulation time

T	measurement transformation matrix
T_{11}	measurement transformation matrix of system state
T_{22}	measurement transformation matrix state derivative
u	input to 1-DOF system
u_d	input realising the desired trajectory
$\underline{u}$	input vector to 3-DOF system
u_c	control force
u_e	excitation force
$\underline{v}$	measurement noise vector
$\underline{w}$	system disturbances vector
$\underline{x}$	system state vector
$\underline{x}_d$	desired trajectory
$\underline{y}$	measurement vector

Abbreviations

CRCTC	Computed Reference Computed Torque Controller
DOF	Degree of Freedom
DOFs	Degrees of Freedom
MATLAB	Matrix Laboratory
SCTC	Sliding Computed Torque Controller
SMC	Sliding Mode Controller
SMO	Sliding Mode Observer
ZMWN	Zero Mean White Noise

Chapter One

Introduction

The first section of this chapter will briefly describe the objective and the assignment of this internship research-project. Then, a small historical overview of the research on the control of vibrations conducted at the Eindhoven University of Technology will be given. In the second section of this chapter the physical dimensions of the beam system will be presented. Finally the current status of the research program will be treated.

1.1 Backgrounds

The Research-Project

A multi degree of freedom (m-DOF) beam system with one-sided spring exhibits a very different response to excitation in comparison with a beam system without such a local nonlinearity. A main difference is the existence of multiple solutions for certain values of a design parameter, the excitation frequency, for example. This feature enables the choice of the most suitable solution, the one of littler amplitude, for instance. The change between solutions can be effectuated by means of control. The accuracy of the tracking behaviour of this controller depends on the accuracy of the estimated state variables.

Assignment

As is explained in the foregoing the nonlinear character of the beam system causes the coexistence of two or more solutions of the system's dynamic equations in steady-state response. The objective is to reduce the amplitude of the steady-state vibration by forcing the system into the natural vibration with the smallest amplitude. Due to the fact that this solution of little amplitude is a natural solution of the ucontrolled system only a small control effort will be needed once the control objective is achieved. This report will describe the research conducted on the assignment to design a sliding observer (Slotine, Hedrick & Misawa [13]) for the nonlinear beam system with one-sided spring and compare its performance (accuracy of the state estimation and required calculation time) with the currently used state reconstruction.

Justification

Many mechanical systems are subject to vibrations due to dynamic excitation. In engineering practice large amplitude vibrations in harmonically excited nonlinear dynamic systems are frequently met, for instance gear rattle or ships colliding at quay sides. Large amplitude vibrations, in particular, are undesired because of the stresses, strains, wear and loud noises they indulge. Since stress and strain may cause defections to the mechanical system, investigating ways to suppress or eliminate these vibrations is justified.

Overview

The fundaments of research into the control of vibrating dynamic systems at the Eindhoven University of Technology were laid in January 1992 (R.H.B. Fey, [3]), by a dissertation on the steady-state behaviour of dynamic systems with local non-nonlinearities. The long-term behaviour of such a system under periodic excitation may be periodic, quasi-periodic, or chaotic. In case of periodic behaviour, the frequency of the system may be equal to the excitation frequency (the harmonic solution) or some constant factor times that frequency. When the frequency of the response equals n times the excitation frequency the n (super)harmonic solution is found; when the response-frequency equals $1/n$ times the excitation frequency it is the $1/n$ (sub) harmonic response ($n \in \aleph, n \geq 2$). Often an unstable harmonic solution of a much smaller maximum displacement (in absolute sense) coexists with a subharmonic solution. An example of a system showing subharmonic behaviour is the system under consideration: a two-dimensional harmonically excited beam system supported by a one-sided spring in its midpoint. Figure 1.1 shows the experimental response of the uncontrolled system to various excitation frequencies together with the analytical solutions for those frequencies. Note that the experiment ('+' and 'o') only yields large amplitude (stable) oscillation while the analytical solution ('.') also contains the unstable, lower amplitude, responses.

The mentioned beam system was built to complement the theoretical analysis with some experimental results. Since then several other graduate and PhD students of the Department of Mechanical Engineering at the Eindhoven University of Technology have researched the dynamic behaviour of this system with a local-nonlinearity. Here, a reference is made to their reports for more information on the dynamical analysis and experimental results: F. Assinck [1], M. de Goeij [4], L.T.A. Sanders [12], E.L.B. van de Vorst [15], and T. de Vries [16].

1.2 The Beam System with One-Sided Spring

In this section the system under consideration will be introduced

Physical Dimensions and Measurements

The beam system under consideration (figure 1.2) consists of a steel beam supported at both ends by two leaf springs. Leaf springs have a large stiffness in longitudinal direction and low stiffness in transversal direction. The one-sided spring is constructed by a clamped beam parallel to the main beam. For positive midpoint displacements ($y_m(t) > 0$) the beam system encounters some additional stiffness.
Harmonic excitation is realised by means of a rotating mass unbalance attached to the middle of the main beam. The mass unbalance is driven by a tacho-controlled motor via a flexible shaft. The shaft has practically no stiffness in transversal direction which leads to a free motion of the main beam in this direction. To avoid phase lag between the desired and realised harmonic excitation the shaft is

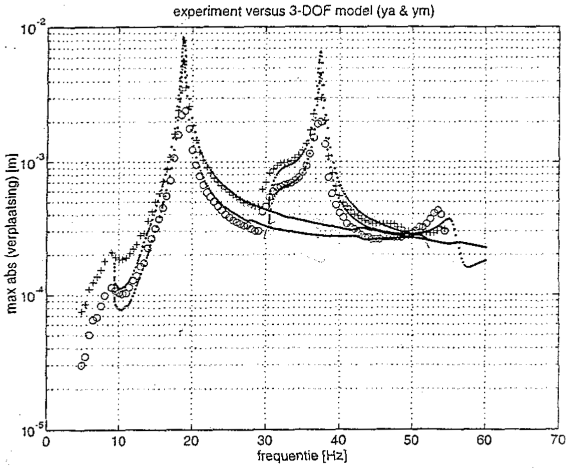


Figure 1.1 Stable and Unstable Solutions of the Beam System

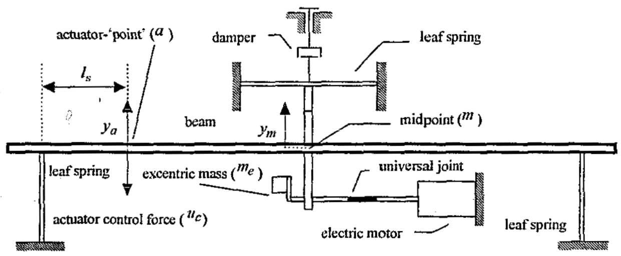


Figure 1.2 The Beam System with one-sided Spring

sufficiently stiff in rotational direction. The control force is applied to the main beam by an actuator at some distance (l_s) from the leaf spring. The actuator translates a current through a coil into a force on the beam. Sensors are mounted on the beam at its midpoint and at the actuator's point of attachment measuring the displacements and accelerations of these two points.

Degrees of Freedom

The beam is described mathematically by three DOFs - two physical and one virtual. The actuator and midpoint displacement (y_a and y_m, respectively) are the physical DOFs, extra DOFs can be included for an accurate description of each oscillation mode of the beam; they have no physical connotation. The DOFs column of the beam system ($\underline{q} = [y_a \quad y_m \quad \xi]^T$) contains one virtual DOF (ξ) representing the first eigenmode of the system (Kant [7]).

1.3 Objective

The recently attained results of the project will be discussed briefly followed by the intentions of this report.

Current Status

Vibration reduction is realised in both simulation and experiment by forcing the system at a prescribed excitation frequency from the stable *1/2* subharmonic response of large amplitude towards the unstable solution of equal frequency but smaller amplitude using a Sliding Computed Torque Controller (SCTC), (Heertjes *et al.* [5]). Algebraic equations are being solved to reconstruct the system's full state especially the modal DOF: ξ. The system equation provides three coupled differential equations (for each DOF: $\ddot{\underline{q}} = [\ddot{y}_a \quad \ddot{y}_m \quad \ddot{\xi}]^T$) in nine unknowns: $\ddot{\underline{q}}$, $\dot{\underline{q}}$ and $\underline{q}$. The variables $\ddot{y}_a, \ddot{y}_m, y_a$, and y_m are being measured, leaving five unknowns. Using an Euler based differentiation scheme the velocities $\dot{y}_a$ and $\dot{y}_m$ are reconstructed. An algebraically solvable system of three equations in three unknowns then remains. The advantage of this simple state reconstruction is its limited required calculation time which is important in experimental set-up. However, poor state reconstruction leads to poor tracking behaviour. Even more important for future research: this reconstruction algorithm cannot be used in models with more modal DOFs using the same set of measured variables. Expanding the system with more than one modal DOF would leave an unsolvable system.

Various experimental runs with the controlled beam system using a SCTC to force the beam to its unstable $1/2$ subharmonic solution have been carried out by Sanders [12]. As described by Heertjes *et al.* [5] other control strategies should be investigated in order to attain the effect of a diminishing necessary control force. Currently the implementation of a Computed Reference Computed Torque Controller (CRCTC) and its possible advantages, as Sanders mentioned in his recommendations, are being scrutinised. In his report [12] Sanders also stated the results obtained in simulation are somewhat misleading because of the discrepancy between (seemingly) continuous time simulation and experiments with constant time steps (discrete-time signals). Therefore, while researching methods for a better reconstruction of the system state, special attention will be paid to the discrete character of the simulation signals.

Chapter Two

Sliding Motion

The goal when observing a system is similar to the one when controlling. Instead of minimising a tracking error it is tried to attain a minimal estimation error. The tracking error ($\underline{e}_x$) then becomes the estimation error vector denoted by ($\underline{e}_{\hat{x}}$). The most important problem we encounter when implementing this concept in an observer structure is the lack of the total state $\underline{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$; often only x_1 is being measured. In this chapter the theoretical background of the sliding observer will be explained using a simple second order 1-DOF system.

2.1 A Simple Example

The following will, very briefly, describe the basics of the sliding surface theory [6].

Sliding Mode Control

In order to show the parallels between a sliding mode controller (SMC) and a sliding mode observer (SMO) a 1-DOF second order mass-spring-damper-system is introduced ($\dot{\underline{x}} = A\underline{x} + \underline{b}u$). The objective when controlling a system usually is to minimise a certain tracking error ($\underline{e}_x$) defined as the difference between the system state ($\underline{x}$) and a predefined desired trajectory ($\underline{x}_d$). Assuming the trajectory can be realised an input variable u_d must exist such that

$$\dot{\underline{x}}_d = A\underline{x}_d + \underline{b}u_d = \begin{bmatrix} 0 & 1 \\ -k/m & -d/m \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u_d \qquad (1),$$

introducing the system-matrices. The tracking error ($\underline{e}_x = \underline{x}_d - \underline{x}$) equation then becomes

$$\dot{\underline{e}}_x = \dot{\underline{x}}_d - A\underline{x} - \underline{b}u = A\underline{e}_x + \underline{b}(u_d - u) \qquad (2),$$

given some initial error $\underline{e}_x(t_0) = \underline{x}_d(t_0) - \underline{x}(t_0)$. The tracking error can be diminished by feeding back $e_{x2} = -\lambda e_{x1}$; λ being a positive constant. This is not always possible since the initial state may not satisfy this equation. Therefore an extra variable (s) is added to compensate for the initial difference

$$e_{x2} = -\lambda e_{x1} + s \qquad (3),$$

defining a transformation. Using feedback linearisation the controlled input incorporating this transformation needs to be

$$u = u_s + u_d + (-\lambda^2 + \lambda \tfrac{d}{m} - \tfrac{k}{m}) e_{x1} + (\lambda - \tfrac{d}{m}) s \qquad (4),$$

forcing s to 0 by choosing the input u_s. Subsequently the tracking error will decrease to zero according to

$$e_{x1}(t) = e_{x1}(t_a) e^{-\lambda(t-t_a)} \quad \forall\, t \geq t_a \qquad (5),$$

with t_a being the point in time at which the *sliding surface* is attained. The sliding surface is the surface $s = 0$, along which the tracking error will slide to $\underline{e}_x = \underline{0}$. The point is a globally and asymptotically stable solution of the differential equation $\dot{e}_{x1} + \lambda e_{x1} = 0$; meaning that if e_{x1} is zero, e_{x2} will become zero as well; for $t \to \infty$. Although the equation ($e_{x2} + \lambda e_{x1} = 0$) actually defines a line on the error surface it is referred to as a sliding surface in the error space as is common practice. One usually deals with higher order or more DOF-systems whose error spaces indeed are of larger dimension containing sliding surfaces of some lower dimension.)

The renewed objective now is to find an expression for u_s in order to attain sliding motion. Using the theory of Lyapunov regarding the stability of solutions [6] the following expression is chosen

$$u_s = k_s \, sign(s) \qquad (6)$$

Although the switching surface theory is outlined here for a scalar tracking error it can be applied just as easily to the vectorial case. For a more detailed outline is referred to [6]. Sliding surfaces have been investigated mostly in the Soviet literature where they were used to stabilise a class of nonlinear systems. Although a sliding mode controller (SMC) features excellent characteristics as robustness in the face of parameter uncertainty, a classical SMC exhibits some severe drawbacks. The most important of these is chattering which includes high control activity and large control authority. Chattering may excite the (usually unmodelled) higher order vibration modes which limits the practical applicability of a SMC. These problems can be remedied by introducing a control smoothening boundary layer around the sliding surface. The boundary layer is defined by

$$sat(\beta) = \begin{cases} 1 & if\ \beta > 1 \\ \beta & if\ -1 \leq \beta \leq 1 \\ -1 & if\ \beta < -1 \end{cases} .$$

The Sliding Observer

Just as the sliding controller consists of a linear part complemented by some nonlinear ($sign$-)term to compensate for errors the sliding mode observer incorporates a linear estimator with nonlinear terms added to it. The linear part of the state estimation is done by a full order 'identical' Luenberger observer, its poles slightly faster than those of the linear part of the observed system [9, 10]. To analyse the performance of the sliding observer the 1-DOF (y_m) model of the nonlinear beam (figure 1.2) with a single measurement is considered:

$$\begin{aligned}\dot{\underline{x}} &= A_{nl}(\underline{x})\underline{x}+\underline{b}\,u=\begin{bmatrix}0 & 1\\ -\dfrac{k+k_{nl}(y_m)}{m} & -{}^{d}\!/_{m}\end{bmatrix}\underline{x}+\underline{b}\,u\\ &= A\underline{x}+\underline{b}\,u+\Delta A_{nl}(\underline{x})\underline{x}=\begin{bmatrix}0 & 1\\ -{}^{k}\!/_{m} & -{}^{d}\!/_{m}\end{bmatrix}\underline{x}+\underline{b}\,u+\begin{bmatrix}0 & 0\\ -\dfrac{k_{nl}(x_1)}{m} & 0\end{bmatrix}\underline{x}\end{aligned} \qquad (7),$$

$$y=\underline{c}^T\underline{x} \qquad (8).$$

The added nonlinearity is described mathematically by

$$k_{nl}(y_m)=\begin{cases}0 & \text{if } y_m<0\\ k_{ls} & \text{if } y_m\geq 0\end{cases} \qquad (9),$$

with k_{ls} the stiffness of the one-sided spring. Defining the estimation error by $\underline{e}_{\hat{x}}=\underline{x}-\hat{\underline{x}}$ the observer is defined as

$$\dot{\hat{\underline{x}}}=A\underline{x}+\underline{b}\,u+\underline{k}_{lo}(y-\hat{y})+\underline{k}_{smo}sign(y-\hat{y}) \qquad (10),$$

with $\underline{k}_{lo}$ being the Luenberger observer gain column and $\underline{k}_{smo}$ the sliding mode observer gain column. Note that the matrix A is in the observer equation implying exact knowledge of the linear system.

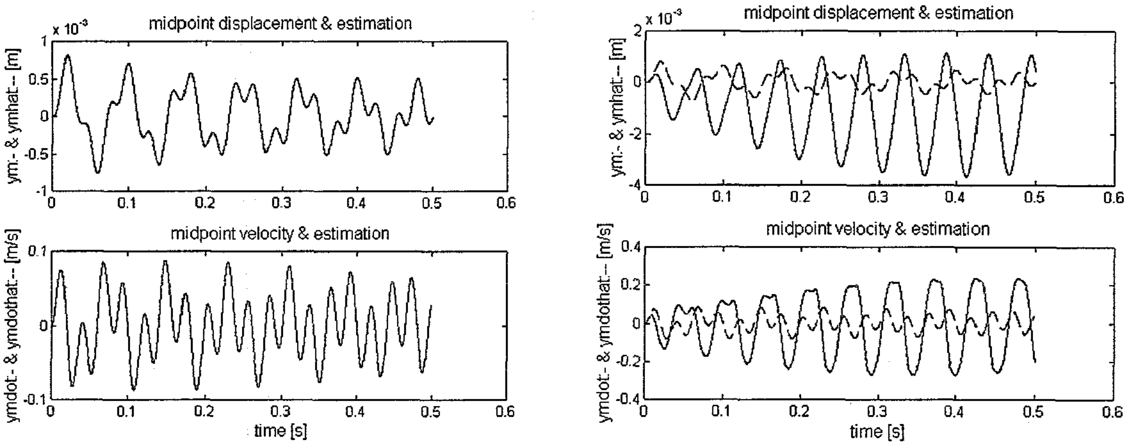


Figure 2.1 Accuracy of the estimation by a Luenberger observer; estimating the state of linear (left) and of a nonlinear system (right).

Figure 2.1 shows a Luenberger observer might suffice for the observation of a linear system but in the case of nonlinear system it looses track of the state. Figure 2.2 shows the performance of the sliding observer in case of a nonlinear system and an initial estimation error. In order to explain how this result was achieved, first the system function is defined

$$\begin{aligned}\dot{x}_2 &= \ddot{x}=\mathrm{X}_{nl}(\underline{x},u)=\mathrm{X}(\underline{x},u)+\Delta\mathrm{X}_{nl}(\underline{x})\\ &=(-{}^{k}\!/_{m}x_1-{}^{d}\!/_{m}x_2+{}^{1}\!/_{m}u)+(-\frac{k_{nl}(x_1)}{m}x_2)\end{aligned} \qquad (11),$$

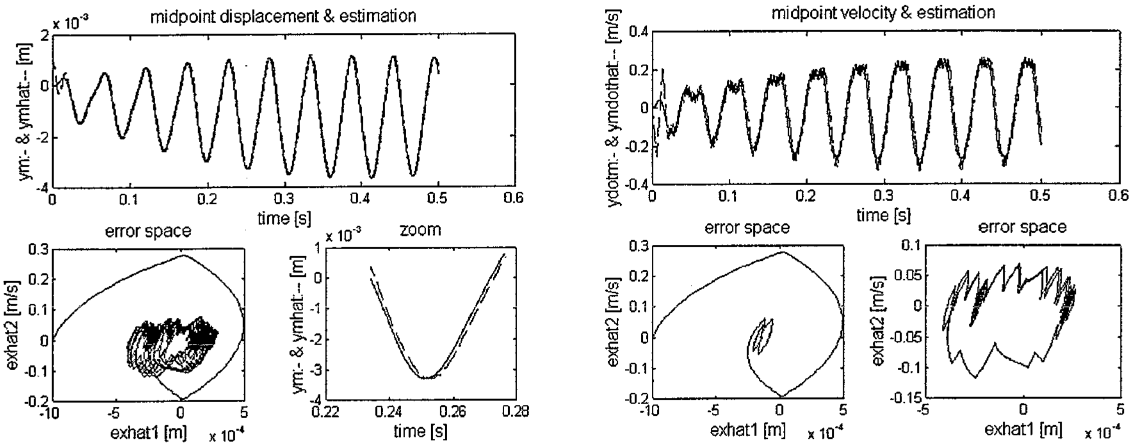


Figure 2.2 The performance of a sliding observer; displacement and velocity (upper right and left), the total error space (lower left), zooming in on the estimation (lower second to left), the error space in the beginning (lower second to right), and another portion of the error space (lower right plot).

and the estimation error 'system-matrix' as

$$A_e = A - \underline{k}_{lo}\underline{c}^T \qquad (12).$$

The estimation error then becomes

$$\dot{\underline{e}}_{\hat{x}} = A_e \underline{e}_{\hat{x}} + \Delta A_{nl}(\underline{x})\underline{x} - \underline{k}_{smo} sign(e_{x1}) \qquad (13).$$

The lower second to right plot of figure 2.2 shows sliding behaviour is generated in the region $|e_{\hat{x}2}| < k_{smo1}$, $e_{\hat{x}1} = 0$ which is referred to as the *sliding patch*. Using Lyapunov's Second Method it can be shown that this region is attractive. Thereto equation 13 is expanded

$$\dot{\underline{e}}_{\hat{x}} = \begin{bmatrix} A_{e11}e_{\hat{x}1} + A_{e12}e_{\hat{x}2} - k_{smo1} sign(e_{\hat{x}1}) \\ A_{e21}e_{\hat{x}1} + A_{e22}e_{\hat{x}2} + \Delta X_{nl}(\underline{x}) - k_{smo2} sign(e_{\hat{x}1}) \end{bmatrix} \qquad (14),$$

verify that $A_{e11} < 0$ and (in case of a single measurement) $A_{e12} = 1$. Next, a candidate Lyapunov function is defined

$$V(\underline{e}_{\hat{x}}, t) = \tfrac{1}{2} e_{\hat{x}1}^2 \quad \Rightarrow \quad \dot{V}(e_{\hat{x}}, t) = e_{x1}\dot{e}_{x1} \qquad (15).$$

applying the condition for global stability, the area of attraction to the sliding patch is found

$$\dot{V}(\underline{e}_{\hat{x}}, t) < 0 \quad \Rightarrow \quad \begin{cases} e_{\hat{x}2} < k_{smo1} + |A_{e11}| e_{\hat{x}1} & if\ e_{\hat{x}1} > 0 \\ e_{\hat{x}2} > -k_{smo1} + |A_{e11}| e_{\hat{x}1} & if\ e_{\hat{x}1} < 0 \end{cases} \qquad (16).$$

The nonlinear parameter k_{smo1} not only defines the switching in the first observer equation but also the size of the sliding patch and therefore the maximum velocity estimation error. The dynamic behaviour on the patch can be analysed using Filipov's solution concept [13] formalizing engineering intuition: the dynamics on the patch can only be a convex combination of the dynamics on each side of the patch

$$\dot{e}_{x2} = \Delta X_{nl}(\underline{x}) + (A_{e22} - \frac{k_{smo2}}{k_{smo1}}) e_{\hat{x}2} \qquad (17).$$

The value of k_{smo2} should be chosen such that it ensures the decline of $e_{\hat{x}2}$. A fair suggestion therefore is $k_{smo2} \geq |\Delta X_{nl}|$. The observer then not only compensates for the system nonlinearities but many times over compensates for them in which case the estimation will converge to the right value (equation 17 will then be negative).

Note the equivalence between the sliding patch and the previously discussed sliding surface. Both are attractive regions and in both cases the error declines as long as it remains in that region. In observer theory an error surface cannot be defined however since such a definition would require the total state to be known which, as was mentioned earlier, is typically not the case when observing.

Chapter Three

Observing the 3-DOF System

The 1-DOF system of the previous chapter is expanded and various methods for reconstruction of its state are treated.

3.1 Model Synthesis

The 3-DOF model used to describe the nonlinear beam system is derived from a Finite Element Method (FEM) model. Several students have devoted thesis projects and internship assignments to attain a valid few DOFs model of the beam system. The construction of the 3-DOF model will be described shortly.

Finite Elements

The FEM-model describing the linear beam was reduced to three DOFs based on the first eigenmode and two residual flexibility modes. For these DOFs the mass-, damping-, and stiffness matrix were derived [9]. The local nonlinearity (one-sided spring) was added to the linear model after reduction. Since the number of DOFs equals the number of modes that are incorporated the validity bandwidth of the model is limited. The reports by Robert Kant and Borre Sanders [7, 12] will give more insight on how the 3-DOF model was attained and its validity limitations. In this internship project the model, as it is described here, was used to calculate the dynamic behaviour of the beam system with one-sided spring.

3-DOF Model

The nonlinear model equation is

$$M\ddot{\underline{q}} + D\dot{\underline{q}} + K_{nl}(\underline{q})\underline{q} = H\underline{u} \qquad (18).$$

Here M, D, and $K_{nl}(\underline{q})$ are the system matrices. As was mentioned in the previous paragraph the one-sided spring is added to the linear model equation; mathematically it is described by

$$K_{nl}(\underline{q}) = K + \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{nl}(y_m) & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (19).$$

Vector $\underline{u}$ consists of both the input signals to the beam system: the control force (u_c) and the excitation force (u_e)

$$\underline{u} = \begin{bmatrix} u_c \\ u_e \end{bmatrix} = \begin{bmatrix} u_c \\ m_e r_e \omega_e^{2} \cos(\omega_e t) \end{bmatrix} \qquad (20).$$

The matrix H makes sure the input signals are applied to the right DOF

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad (21).$$

System Function

The 3-DOF model (18) then is rearranged to

$$\underline{\ddot{q}} = -M^{-1} K_{nl}(\underline{q})\underline{q} - M^{-1} D\underline{\dot{q}} + M^{-1} H\underline{u} \qquad (22).$$

It is assumed that system disturbances are modelled quite reasonably by a Zero Mean White Noise (ZMWN) signal ($\underline{w} = \begin{bmatrix} \underline{w}_1^T & \underline{w}_2^T \end{bmatrix}^T$).

$$\underline{\ddot{q}} = \underline{X}(\underline{\dot{q}}, \underline{q}, \underline{u}, \underline{w}_2) = -M^{-1} K_{nl}(\underline{q})\underline{q} - M^{-1} D\underline{\dot{q}} + M^{-1} H\underline{u} + \underline{w}_2 \qquad (23).$$

State Description

In order to make the model suitable for control theory simulating purposes the system function is written in companion form. The state vector is defined as

$$\underline{x} = \begin{bmatrix} y_a & y_m & \xi & \dot{y}_a & \dot{y}_m & \dot{\xi} \end{bmatrix}^T = \begin{bmatrix} \underline{q}^T & \underline{\dot{q}}^T \end{bmatrix}^T = \begin{bmatrix} \underline{x}_1^T & \underline{x}_2^T \end{bmatrix}^T \qquad (24),$$

$$\underline{\dot{x}} = \begin{bmatrix} \underline{\dot{x}}_1 \\ \underline{\dot{x}}_2 \end{bmatrix} = \begin{bmatrix} \underline{x}_2 + \underline{w}_1 \\ \underline{X}(\underline{x}, \underline{u}, \underline{w}_2) \end{bmatrix} = A_{nl}(\underline{x})\underline{x} + B\underline{u} + \underline{w} \qquad (25).$$

the state description matrices then become

$$A_{nl}(\underline{x}) = \begin{bmatrix} 0 & I \\ -M^{-1} K_{nl}(\underline{q}) & -M^{-1} D \end{bmatrix} \qquad (26),$$

$$B = \begin{bmatrix} 0 \\ M^{-1} H \end{bmatrix} \qquad (27).$$

The linear system matrix is denoted by A, and the difference between the linear and nonlinear system matrix $\Delta A_{nl}(\underline{x}) = A_{nl}(\underline{x}) - A$. The linear system matrix is

$$A = \begin{bmatrix} 0 & I \\ -M^{-1} K & -M^{-1} D \end{bmatrix} \qquad (28).$$

Note that the difference is a function of the system state.

3.2 The Observers

For reconstruction of the system state four state variables are available through measurement.

Measurement

The beam actuator and midpoint displacement are being measured just as their acceleration: $\underline{y}(t)=\left[x_1(t)\quad x_2(t)\quad \dot{x}_4(t)\quad \dot{x}_5(t)\right]^T$. To incorporate in the model the measurement noise that is encountered in the experimental set-up, it is assumed that the noise signals are described reasonbly well by a ZMWN signal. The ZMWN measurement noise vector will be denoted by $\underline{v}(t)$. The transformation matrix (T) connecting the measured DOF-column with the DOF-vector is defined by $\underline{q}_m = T\underline{q}$ and is the same for the acceleration measurement: $\underline{\ddot{q}}_m = T\underline{\ddot{q}}$. The matrices transforming the state vector are then defined as $\underline{y} = T_{11}\underline{x} + T_{22}\underline{\dot{x}} + \underline{v}$ with $T_{11} = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}$ and $T_{22} = \begin{bmatrix} 0 & 0 \\ 0 & T \end{bmatrix}$.

Algebraic Reconstruction

Since implementation in the experimental set-up required a method using little calculation time Borre Sanders [12] developed a straightforward but useful method for reconstructing the system state, the modal DOFs to be more specific: solving the equation algebraically. Observing the full state of the 3-DOF system algebraically means solving three equations in nine unknowns: $\underline{q}$, $\underline{\dot{q}}$ and $\underline{\ddot{q}}$. The velocities are reconstructed according to

$$\begin{bmatrix} x_4(t) \\ x_5(t) \end{bmatrix} = \left(\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} - \begin{bmatrix} x_1(t-T_s) \\ x_2(t-T_s) \end{bmatrix}\right) + \frac{T_s}{2}\begin{bmatrix} \dot{x}_4(t-T_s) \\ \dot{x}_5(t-T_s) \end{bmatrix} \qquad (29).$$

A solvable system of three equations in three unknowns remains. For the exact equations is referred to the report by Borre Sanders [12].

Although the algebraic state reconstructor seems very attractive because of its simplicity it conceals some disadvantages. For instance, when rearranging the fourth and fifth system equation the influence of the state variables that are to be estimated (x_3 and x_4) is neglected. But, more important, the reconstructor will not support expansion of the system with more modal DOFs: it will leave a set of equations in too many unknowns. Using this reconstruction algorithm for the same system expanded with one modal DOF would yield a set of four equations in six unknowns

Observability

As discussed in the previous chapter the basis of the state estimation by the sliding observer is done by a conventional Luenberger observer. The nonlinear extensions are designed to react quickly to the deviations of the linear estimation from the nonlinear system or to compensate for noise and disturbances. If it is assumed that the nonlinear extensions of the linear estimator indeed compensate for the added system-nonlinearity it is plausible that a necessary condition for a reasonable reconstruction of the state is the pair (A,C) to be detectable (Kok [8], Walcott & Zak [17]), which indeed it is. The same condition needs to be verified for a Kalman filter to be able to observe the linear system.

Kalman Filter

In order to judge the possible improvements in the state estimation brought about by the sliding mode observer its performance is compared not only to the algebraic reconstructor but to a Kalman filter as well. The optimal Kalman filter gain matrix is calculated using weight matrices that express the correctness of the various measurements and system equations. By adjusting the values of the weight matrices a linear Kalman filter can be applied for estimation of the nonlinear system (Sorenson [14]). In doing so quite a reasonable state estimation can be attained due to the fact that during a large part of its oscillation the system indeed behaves as a linear one; as was concluded earlier by Robert Kant [7] and Edward van de Vorst [15]. Extensive backgrounds on a broad variety of aspects of Kalman filtering can be found in the publication by Sorenson [14].

The Sliding Mode Observer

As was discussed in the previous chapter, the SMO is designed to react to estimation errors according to the following scheme. Quite large initial errors will be diminished proportionally by the Luenberger part of the observer complemented by the nonlinear terms. When the error is of an order comparable to that of the switching terms the Luenberger gains will have become relatively small. The objective then is to bring the zeroth order variable of each DOF (the displacements, $\underline{q}$) to zero (which is equivalent to reaching the sliding patch) using the switching terms. If the sliding patch is reached the higher order state variables will become zero as well. The fundamental quality of a sliding observer lies herein that providing some knowledge about the norm of the dynamic uncertainty its effect can be compensated.

Mathematically

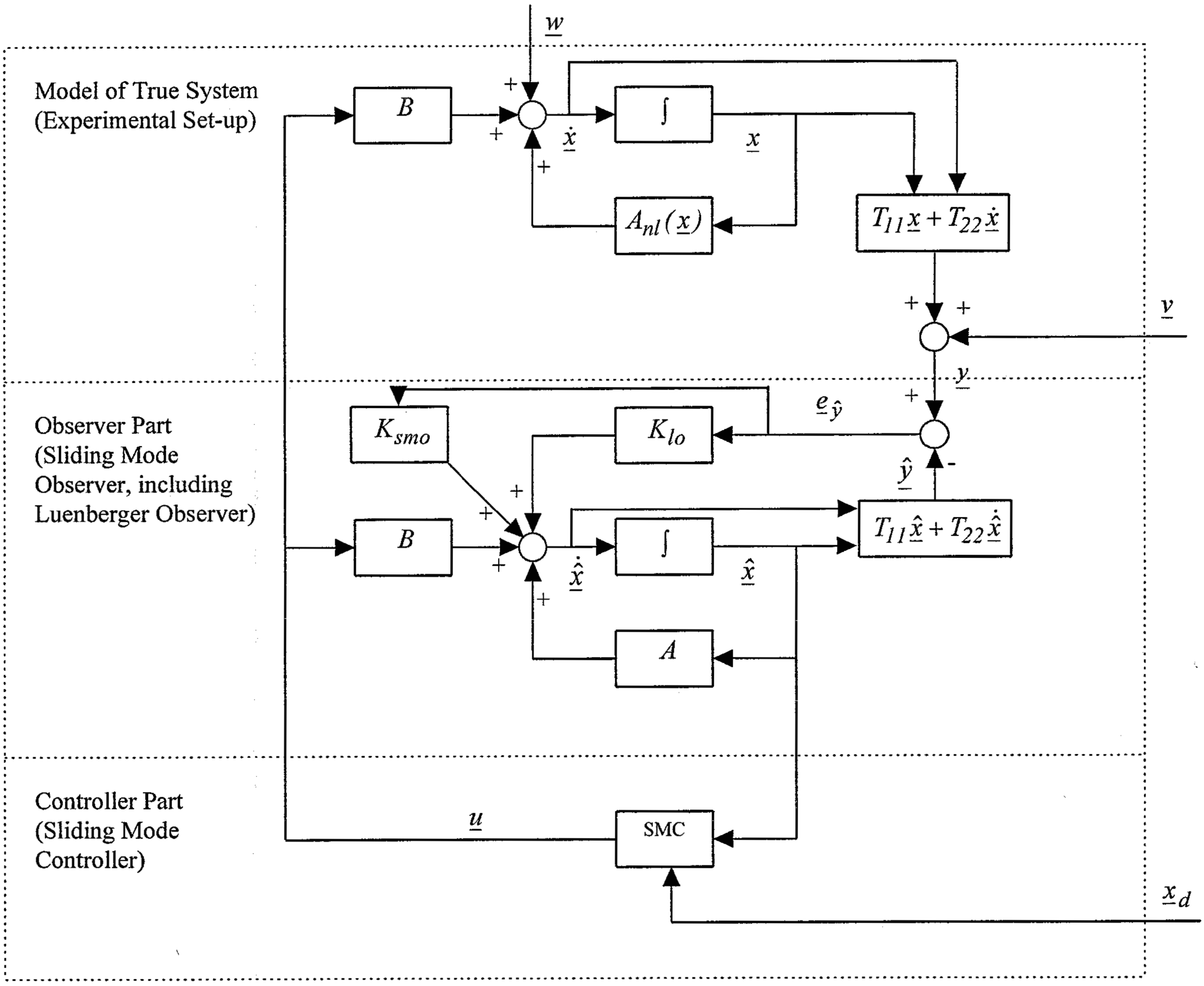
The mathematical equations of the sliding observer can be found a simple expansion of the scalar set of observer equations (10)

$$\dot{\hat{\underline{x}}} = A\hat{\underline{x}} + B\underline{u} - K_{lo}(\underline{y} - \hat{\underline{y}}) - K_{smo} sign(\underline{y} - \hat{\underline{y}}) \qquad (30),$$

with the nonlinear switching-terms matrix being

$$K_{smo} = \begin{bmatrix} k_{smo1} & 0 & 0 & 0 \\ 0 & k_{smo2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_{smo3} & 0 & 0 & 0 \\ 0 & k_{smo4} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (31).$$

Note that only the measurement of the displacements of each DOF is used to update the estimation since in the sliding mode observer theory the nonlinear gain terms only use the displacement of the estimation error as their argument. The crucial point in applying the SMO to the nonlinear beam system is that for modal DOFs no measurement exists and therefore the modal DOF estimation error is not available to the observer making it impossible to adapt the estimation of this variable with accurate information.



$\underline{w}$
Model of True System (Experimental Set-up)
B
+
+
+
$\dot{\underline{x}}$
$\int$
$\underline{x}$
$A_{nl}(\underline{x})$
$T_{11}\underline{x} + T_{22}\dot{\underline{x}}$
+
+
$\underline{v}$
$\underline{y}$
Observer Part (Sliding Mode Observer, including Luenberger Observer)
K_{smo}
K_{lo}
$\underline{e}_{\hat{y}}$
+
-
$\hat{\underline{y}}$
+
+
+
+
B
$\dot{\hat{\underline{x}}}$
$\int$
$\hat{\underline{x}}$
$T_{11}\hat{\underline{x}} + T_{22}\dot{\hat{\underline{x}}}$
A
Controller Part (Sliding Mode Controller)
$\underline{u}$
SMC
$\underline{x}_d$

Figure 3.1 Block Diagram of the Controlled Beam System

Chapter Four

Simulations

The sliding mode state reconstructor that has been outlined so far is analysed in this chapter through simulations in the signal processing package MATLAB.

4.1 Routines and Conditions

The purpose of the various m-files (MATLAB-routines) and simulation environment will be treated briefly. A listing of the m-files can be found in the appendix.

Integration Methods

Although it has caused a reasonable amount of numerical difficulties, quite consistently is being held on to simulation in discrete steps: incorporating the sample period. In his report [12] Borre Sanders mentioned quite a discrepancy between computer simulations and experimental results arises when those simulations use variable step algorithms [2] in order to increase the accuracy of the simulation. Because of the adaptation of integration periods the simulated signals become (semi-)continuous. In experiments however, the signal processing environment is discrete and of constant period, usually the period between two measurements. These considerations do not mean that no variable period integration routines have been used at all. For some purposes it proved to be very useful to simulate seemingly continuous; as was done for the analysis in chapter two.

Simulation Routines

The current experimental sample frequency ($f_s = 500$ [Hz]) was taken and a new measurement was simulated every sample period (T_s) in order to attain valid results. Furthermore, as was shown in the theory of the preceding chapters, to increase the validity the measurement was corrupted by a noise signal ($\underline{v}$) while the system itself also encountered some disturbances $\underline{w}$. The heart of the routines, which can be found in the appendix, is formed by two integration routines running in the main routine `dof3.m`. First the MATLAB routine `odex.m` is an customisation of the MATLAB-routine `ode45.m` (a variable step integration routine) which simulates the experimental set-up, secondly a reconstructor m-file is initiated from the main routine. The reconstructor routines are `kalman.m` or `smo.m`. Modelling has been done in separate functions as much as possible to yield readable routines. All trials were done in the Matrix Laboratory software package: MATLAB, version 4.0; on a PC.

Controller

The simulation trials lasted two seconds; a controller action was added to the system as of $t = 1$ [s]; after the transient had faded. The details of how the sliding mode controller parameters were established will not be discussed in this report since it is not in the scope of the assignment - the basic principles of a sliding controller in general were discussed in chapter two. For details on the performance of the SMC is referred to the reports of Borre Sanders [10], Robert Kant [7] and [6], and to reports to be published since research on optimising the controller performance is being conducted at this very moment. The sliding mode controller parameters will suffice here $\lambda = 100$, $\eta = 25$, and $\sigma = 0.5$.

4.2 Results

It is explained how the observer parameters were found and some simulation trials are discussed.

Observer Parameters

The results were preceded by various simulation trials in which it has been tried to optimise the accuracy of the estimation. The sliding observer parameters (K_{smo} and $\underline{\varphi}$) were attained in a manner proposed by Slotine, Hedrick & Misawa [13]: The parameters k_{smo1} and k_{smo2} were chosen equal to the desired accuracy in $\hat{x}_4$ and $\hat{x}_5$. Consequently k_{smo4} and k_{smo5} were chosen equal to the dynamic uncertainty. They were decreased somewhat since the nonlinear estimator seemed to react quite vigoursly. For this same reason the *sign* function was substituted by a *sat* (saturation) function introducing a boundary layer ($\underline{\varphi}$) around the sliding patch. According to [11] the boundary layer needed to be larger than or equal to two sample periods times the nonlinear gains:

$$\varphi_i = 2T_s k_{smoi} \qquad (32).$$

Trials

Three possible state estimation methods are reviewed in this section: the algebraic reconstructor, the Kalman filter, and the sliding mode observer. First, the performance of the algebraic reconstructor is showed (figure 4.1). Of the three DOFs involved in the system only the midpoint's and modal DOF's displacement are shown since the actuator displacement is comparable to the midpoint displacement. Note that the maximum control force is limited (to 20 [N]) as Sanders [12] did in his analysis.

The two estimation error plots of figure 4.1 (lower left and lower second to right) show the error for a smaller period than the upper two plots but on the same time scale; the period around the activation of the controller ($t = 1$ [s]) is chosen for this plot. The performance of the algebraic recontstructor is quite disappointing: the estimation error remains large causing the controller not to succeed in diminishing the midpoint excitation while a large control force remains.

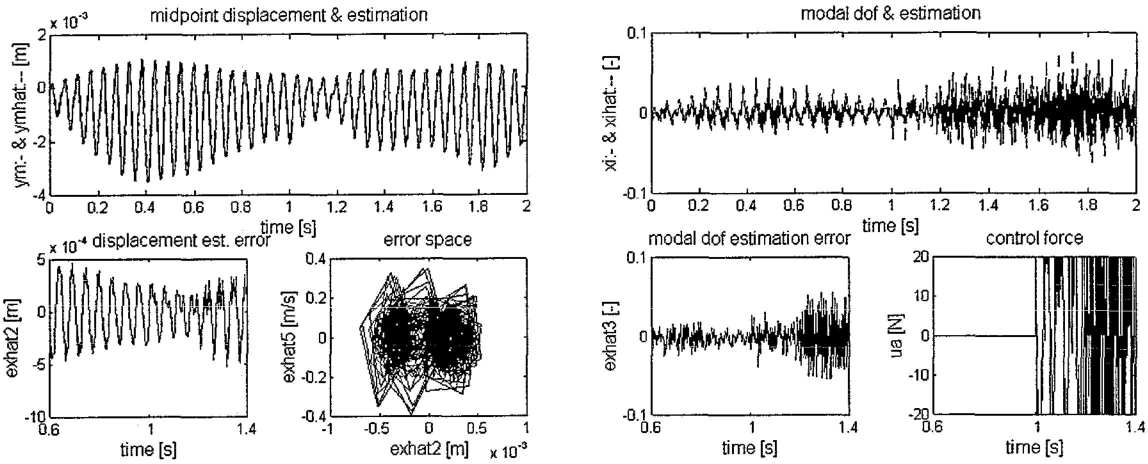


Figure 4.1 Performance of the Algebraic Reconstructor: displacement and estimation of the midpoint (upper left) and modal DOF (upper right), their estimation errors (lower left and lower second to right); the error space (lower second to right) and the applied control force (lower right plot).

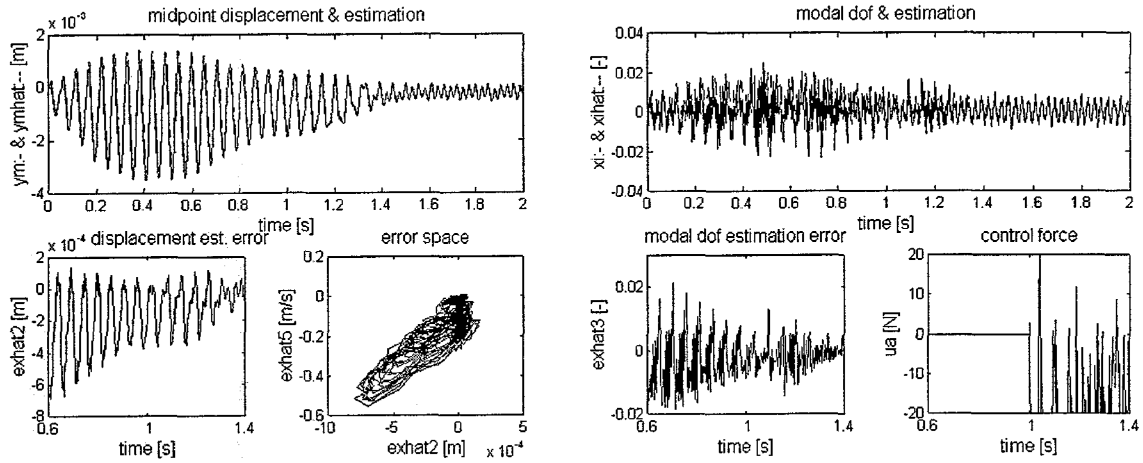


Figure 4.2 Performance of the Kalman Filter: displacement and estimation of the midpoint (upper left) and modal DOF (upper right), their estimation errors (lower left and lower second to right); the error space (lower second to right) and the applied control force (lower right plot).

Applying the Kalman filter for state reconstruction does yield a diminishing midpoint oscillation (figure 4.2, upper left); note the switching of the control force is diminished also, compared to the algebraic reconstructor. The control force is the force required to have

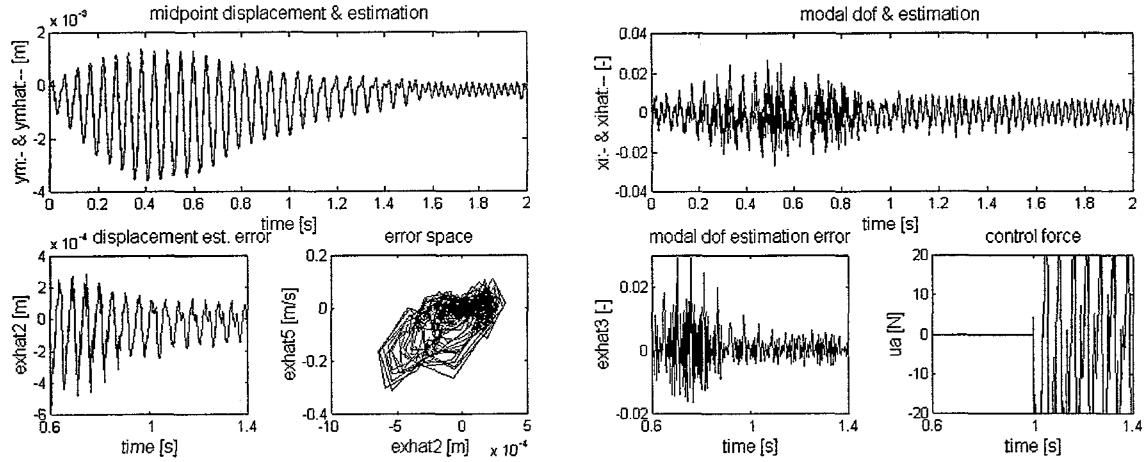


Figure 4.3 Performance of the Sliding Mode Observer: displacement and estimation of the midpoint (upper left) and modal DOF (upper right), their estimation errors (lower left and lower second to right); the error space (lower second to right) and the applied control force (lower right plot).

the beam system oscillate in its unstable solution. Decrement of this force is an important controller design criterion.

Figure 4.3 shows, just as 4.2 does, a diminishing displacement and estimation errors from the moment the controller is switched on. For a more thorough comparison of both reconstructor the relative euclidic norm of both state estimation error vectors is being compared: $\frac{\|\underline{e}_{\hat{x}}\|_2}{\|\underline{x}\|_2} = [0.13 \quad 0.13 \quad 0.98 \quad 0.49 \quad 0.52 \quad 1.36]^T$ for the SMO, and

$$\frac{\|\underline{e}_{\hat{x}}\|_2}{\|\underline{x}\|_2} = [0.13 \quad 0.20 \quad 1.06 \quad 0.74 \quad 1.66 \quad 1.28]^T$$

for the Kalman filter. The SMO performs better overall. Figure 4.4 shows that the nonlinear terms of the sliding mode indeed compensate for the added nonlinearity. The above plot (Kalman filter) shows a significant one-sided error. The Kalman filter only estimates well when the one-sided spring is inactive: the Kalman filter system matrix then equals the true system matrix. The lower plot shows the estimation error of the SMO nearly is a zero mean signal.

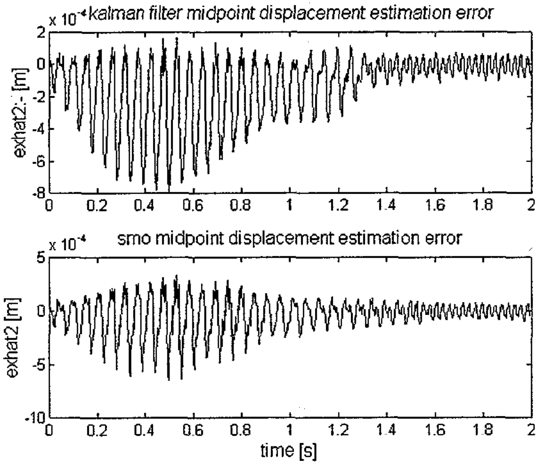


Figure 4.4 Compensation of Nonlinearities

Calculation Time

The performance of a state estimator is not solely based on the accuracy of the state reconstruction but on the period of time it needs to calculate such a reconstruction (T_c) also. Simulation in MATLAB, as in any other signal processing toolbox, takes much longer than an algorithm in machinecode would do. No conclusive values can therefore be retrieved from simulations; however, a comparison is quite useful indeed. The sliding mode observer generally takes twice as long as the other reconstruction algorithms; probably because of the various calculations it involves.

	mean T_c	max T_c
Algebraic reconstructor	0.0076 [s]	0.06 [s]
Kalman filter	0.0074 [s]	0.06 [s]
SMO	0.017 [s]	0.06 [s]

Table 4.1 Calculation Time for various Reconstructor Routines ($f_s = 500$ [Hz]).

Reconsideration

The state reconstruction can be improved fairly well by the reviewed algorithms. However, the results are not quite as good as was hoped for. The reason can partly be found in the fact that the simulated system is corrupted by noise and disturbances, but then, this will be encountered in experiment also. Another important cause is the lack of a measurement of the total first order ($\underline{q}$) as was discussed before. A method to overcome this problem is to use the total first part of the state estimation (including the modal DOF) as an improved measurement and repeat the integration. This algorithm was tried in simulation also the but its results were not significantly better than those of the basic sliding observer and are therefore not shown.

The influence of the simulation period was also analysed. Increasing the simulation frequency (from 500 [Hz] to 1000 [Hz]) did not bring about many visible improvements, the results are therefore judged by comparing the signal norms. The relative norm of the SMO at $f_s = 1000$ [Hz] was $\frac{\|e_{\hat{x}}\|_2}{\|\underline{x}\|_2} = [0.089 \;\; 0.091 \;\; 0.82 \;\; 0.44 \;\; 0.46 \;\; 1.10]^T$ and the norm of its actuator force $\frac{\|u_c\|_2}{n_e} = 0.51$ [N] (n_e being the number of elements in the vector taken from the moment the controller was switched on). Comparing the relative norms of the sliding observer at two simulation frequencies shows the reconstruction is improved somewhat at a larger simulation frequency. Another possible improvement, the replacement of the linear part of the sliding observer by the Kalman filter yielded the following, results: $\frac{\|e_{\hat{x}}\|_2}{\|\underline{x}\|_2} = [0.067 \;\; 0.077 \;\; 0.87 \;\; 0.44 \;\; 0.63 \;\; 1.17]^T$ and $\frac{\|u_c\|_2}{n_e} = 0.44$ [N].

Chapter Five

Conclusions & Recommendations

5.1 Conclusions

The state estimation can indeed be improved by the sliding mode observer. The control force required to persuade the beam system to its unstable solution also diminishes. Simulation trials show that improvement of the algebraic state reconstructor can also be achieved by applying a Kalman filter. Although the state estimation results of the SMO are not a lot better than those of the Kalman filter, comparison of the various norms of the estimated signals show the SMO yields a better estimation. A main obstruction for the SMO to yield more perfect results is the fact that not all DOFs were being measured: the sliding mode observer is designed to update its estimation of a DOF based on a measurement of the displacement of that very DOF which is not possible for the modal DOF. An observer algorithm going over the simulation period twice again did not yield any significant improvement of the estimated state. Ultimately the best result was attained with the SMO having a Kalman filter as linear estimator at $f_s = 1000$ [Hz].

5.2 Recommendations

The best state reconstructor of the analysis showed to be the SMO having a Kalman filter as its linear estimator part at $f_s = 1000$ [Hz]; it is therefore recommended to use this observer in the experimental set-up. It is also recommended to use the largest possible simulation frequency since increasing the simulation frequency has shown to improve the estimation.

The sliding observer seemed a good method for state reconstruction on beforehand. Other (nonlinear) estimators deserve attention also. Of the overview Misawa & Hedrick [11] give in their article the extended Kalman filter probably is the most promising alternative. Especially the constant gain extended Kalman filter which overcomes some computational burden. It is recommended to see whether this observer performs better than the SMO. An important advantage of such a filter is the fact that it is model based and would therefore leave better possibilities for adjusting the estimation of the modal DOF.

Although the discrete time simulation, that has been carried out throughout this report, does impose somewhat of a burden and does not yield the 'smooth' results that one is used to from simulation it is believed that the results comply better with the experimental ones. Therefore it seems advisable to simulate in a truly discrete environment more often.

References

[1] Assinck, F.A.: *Experimentele verificatie van het steady-state gedrag van een opgelegde balk met een lokale niet-lineaire ondersteuning*, WFW-report 93.093, August 1993.

[2] Bosch, P.P.J. van den & A.C. van der Klauw: *Modelling Identification and Simulation of Dynamical Systems*; CRC Press, Baton Rouge, Florida, 1994.

[3] Fey, R.H.B.: *Steady-state behaviour of reduced dynamic systems with local non-linearities*; Ph.D. Thesis, Eindhoven University of Technology, Faculty of Mechnical Engineering, Eindhoven, January 1992.

[4] Goeij, M. de: *Experimenten met de vernieuwde meetopstelling voor de opgelegde balk met lokale niet-lineaire ondersteuning*; Eindhoven University of Technology, WFW-report 94.008, January 1994.

[5] Heertjes, M.F., M.J.G. van de Molengraft, J.J. Kok, R.H.B. Fey & E.L.B. van de Vorst: *Vibration Control of a Nonlinear Beam System*; IUTAM Symposium on Interaction between Dynamics and Control in Advanced Mechanical Systems, 135-142, 1997.

[6] Jager, A.G. de, I. Lammerts & F. Veldpaus: *Course on Advanced Control*; course abstract no. 4668, Eindhoven University of Technology, 1991.

[7] Kant, A.R.: *Stabiliseren van instabiele periodiek oplossingen van niet-lineaire mechanishe systemen*; TNO-report 95-WEC-R0620, WFW-report 59.083, June 1995.

[8] Kok, J.J.: *Werktuigkundige Regeltechniek II*; course abstract no. 4594; Eindhoven University of Technology, 1990.

[9] Luenberger, D.G.: *Observers for Multivariable Systems*; IEEE Transactions on Automatic Control, vol. AC-11, no. 2: 190-197, April 1966.

[10] Luenberger, D.G.: *An Introduction to Observers*; IEEE Transactions on Automatic Control, vol AC-16, no. 6: 596-602, December 1971.

[11] Misawa, E.A. & J.K. Hedrick: *Nonlinear Observers - A State-of-the-Art Survey*; Transactions of the ASME, vol 111: 344-352, September 1989.

[12] Sanders, L.T.A.: *Implemantatie van een schakelvlakregeling op een balk met enkelzijdige veer*, Eindhoven University of Technology, WFW-report 96.033, February 1996.

[13] Slotine, J.-J.E., J.K. Hedrick, & E.A. Misawa: *On Sliding Observers for Nonlinear Systems*; Journal of Dynamic Systems, Measurement, and Control 109: 245-252.

[14] Sorenson, Harold W.: *Kalman Filtering: Theory and Application*; IEEE Press, New York, 1985.

[15] Vorst, E.L.B. van de, *et al.*: *Vibration Control of Periodically Excited Nonlinear Dynamic Multi-DOF Systems*; Journal of Vibration and Control 1: 75-92,1995.

[16] Vries, A.F. de: *Trillingsregeling voor niet-lineaire mechanische systemen: een experimentele analyse*, Eindhoven University of Technology, WFW-report 95.052, April 1995.

[17] Walcott, B.L. & S.H. Zak: *Observation of Dynamical Systems in the Presence of Bounded Nonlinearities/Uncertainties*; Proceedings 25th Conference on Decisision and Control, Athens Greece, December 1996.

Appendix

MATLAB Routines

This appendix will give the programmes and data used in simulation. Simulation was done in MATLAB version 4.0.

3-DOF Basic Routine

```
%---------------------------------------
% dof3.m
%---------------------------------------

clear all

global f k K ts tf u w xd_d x_d

%---------------------------------------
% initialization
%---------------------------------------

dof3par

%---------------------------------------
% initial values
%---------------------------------------

x(:,1)=x0;
xdot(:,1)=xdot0;
xhat(:,1)=xhat0;
xhatdot(:,1)=xhatdot0;
exhat(:,1)=x(:,1)-xhat(:,1);
u(:,1)=u0;
u(:,2)=[u0(1);
        excite(1)];
w(:,1)=w0;
v(:,1)=v0;
ez(:,1)=[0; 0];

%---------------------------------------
% simulation loop
%---------------------------------------

for k=1:(((tf-ts)/Ts)-1)
```

```
  %--------------------------------------
  % estimation error
  %--------------------------------------

  exhat(:,k)=x(:,k)-xhat(:,k);

  %--------------------------------------
  % control force
  %--------------------------------------

  ue=excite(k);

  if ts+(k-1)*Ts>=tc
    [uc,ex(:,k)]=control(xhat(:,k),k);
  else
    uc=0;
  end

  u(:,k)=[uc; ue];

  %--------------------------------------
  % noise
  %--------------------------------------

  %v(:,k)=zeros(4,1);
  %w(:,k)=zeros(6,1);
  v(:,k)=vmax.*randn(4,1);
  w(:,k)=wmax.*randn(6,1);

  %--------------------------------------
  % system
  %--------------------------------------

  [x(:,k+1),xdot(:,k+1)]=odex(ts+(k-
1)*Ts,ts+k*Ts,x(:,k),u(:,k),w(:,k),1e-6,0);

  %--------------------------------------
  % measurement
  %--------------------------------------

  y(:,k)=T11*x(:,k)+T22*xdot(:,k)+v(:,k);

  %--------------------------------------
  % algebraic estimation method
  %--------------------------------------


%[xhat(:,k+1),xhatdot(:,k+1),Tc(k)]=algebra(xhat(:,k),xhatdot(:,k),y
(:,k),u(:,k));

  %--------------------------------------
  % kalman filter estimation
  %---------------------------------------

  [xhat(:,k+1),Tc(k)]=kalman(xhat(:,k),u(:,k),y(:,k));

  %---------------------------------------
  % smo-estimation
  %---------------------------------------

  %[xhat(:,k+1),Tc(k)]=smo(xhat(:,k),u(:,k),y(:,k));

  %---------------------------------------
  % loop end
  %---------------------------------------

end

%---------------------------------------
% corrective vector truncation
%---------------------------------------

x=x(:,1:k);
xhat=xhat(:,1:k);
xdot=xdot(:,1:k);
t=t(1:k);
u=u(:,1:k);
```

3-DOF paramters

```
%---------------------------------------
% dof3par.m
%---------------------------------------

global A B C D eta fe fi H K Ko Ksmo kls M m
          mere S sigma Ts T11 T22 ts tf Vvv Vww we

%---------------------------------------
% initial values
%---------------------------------------

q0=[0; 0; 0];
qdot0=[0; 0; 0];
qddot0=[0; 0; 0];

x0=[q0; qdot0];
xdot0=[qdot0; qddot0];

%xhat0=[1e-3; 1e-3; 1e-2; qdot0];
xhat0=x0;
xhatdot0=xdot0;

u0=zeros(2,1);
```

```
w0=zeros(6,1);
v0=zeros(4,1);

n=size(x0,1);

%----------------------------------------
% system parameters
%----------------------------------------

M=[ 2.559532E+00  -8.761650E-01  -1.015671E-02 ; ...
   -8.761650E-01   4.818900E+00   2.598233E-02 ; ...
   -1.015671E-02   2.598233E-02   2.760633E-04];
K=[ 3.720669E+05  -2.649268E+05  -2.412606E-09 ; ...
   -2.649268E+05   2.219827E+05   4.032910E-09 ; ...
   -2.412606E-09   4.032910E-09   5.814239E+01];
D=[ 1.829682E+00  -1.161981E+00  -6.127086E-03 ; ...
   -1.161981E+00   1.316906E+01   7.155192E-03 ; ...
   -6.127086E-03   7.155192E-03   2.115189E-04];
H=[1 0; 0 1; 0 0];

T=[1 0 0; 0 1 0];
T11=[T zeros(2,3); zeros(2,6)];
T22=[zeros(2,6); zeros(2,3) T];

A=[zeros(n/2) eye(n/2);
   -inv(M)*K -inv(M)*D];
B=[zeros(n/2,2); inv(M)*H];
Bc=[zeros(n/2,1); inv(M)*Hc];

C=[T zeros(2,3); zeros(2,3) T]*...
  [eye(3) zeros(3); -inv(M)*K -inv(M)*D];

fe=37;
we=2*pi*fe;
mere=0.984e-3;
kls=216506;

%----------------------------------------
% time parameters
%----------------------------------------

fs=500;
ts=0;
tf=2;
tc=1;
Ts=1/fs;
t=[ts:Ts:tf];

%---------------------------------------
% observer parameters: sliding mode observer
%---------------------------------------
```

```
fi=[4e-4; 4e-4; 4e-3; 2e-1];

Ksmo=[1e-1 0 0 0;
      0 1e-1 0 0;
      0 0 0 0;
      1e0 0 0 0;
      0 5e1 0 0;
      0 0 0 0];

%--------------------------------------
% noise parameters
%--------------------------------------

vmax=[1e-5; 1e-5; 1e-1; 1e-1];
wmax=[1e-3; 1e-3; 1e-2; 1e-1; 1e-1; 1e0];

%-------------------------------------
% observer parameters: optimal Kalman Filter
%-------------------------------------

Vvv=diag([1e-9 1e-9 1e3 1e3])
Vww=diag([1e-3 1e-3 1e-3 1e-3 1e0 1e-3])

[eigvec,eigval]=eig([A Vww; C'*inv(Vvv)*C -A']);
W=[];
for i=1:12,
  if eigval(i,i)>=0;
    W=[W eigvec(:,i)];
  end
end
W11=W(1:6,:);
W21=W(7:12,:);
Q=real(W11*inv(W21));
Ko=Q*C'*inv(Vvv)
eig(A-Ko*C)
eig(A)

%-------------------------------------
% Fourier-coefficients of desired trajectories
%-------------------------------------
%
%    ya             ym             xi

F=[-9.557878E-05  -1.342321E-04   5.566923E-15 ; ...
   -2.076756E-04  -2.357720E-04  -5.022536E-03 ; ...
    6.593021E-06   2.405319E-06   6.270274E-04 ; ...
   -5.200674E-06   4.360923E-06  -1.061285E-03 ; ...
   -4.726419E-07   1.114107E-06  -1.615968E-04 ; ...
   -4.275982E-08   1.741048E-07  -2.127516E-05 ; ...
    3.469776E-08  -2.037990E-07   2.294010E-05 ; ...
   -1.088744E-08   8.355577E-08  -8.978374E-06 ; ...
   -2.875386E-10   2.705571E-09  -2.824125E-07 ; ...
```

```
     2.718009E-09  -3.001759E-08   3.072926E-06 ; ...
    -1.658489E-09   2.081549E-08  -2.101968E-06 ; ...
     2.682362E-06   3.111014E-06   5.906293E-05 ; ...
    -1.941479E-07  -8.012490E-08  -1.763401E-05 ; ...
     1.282952E-07  -8.792607E-08   2.439795E-05 ; ...
     3.764471E-08  -8.514441E-08   1.254474E-05 ; ...
     2.889259E-09  -1.122814E-08   1.388881E-06 ; ...
    -3.021123E-09   1.711162E-08  -1.939903E-06 ; ...
     1.116409E-09  -8.304462E-09   8.967320E-07 ; ...
     1.377979E-11  -1.211392E-10   1.276046E-08 ; ...
    -3.371688E-10   3.627980E-09  -3.725066E-07 ; ...
     2.332053E-10  -2.860788E-09   2.895584E-07];

%-------------------------------------
% sliding mode control parameters
%-------------------------------------

lambda=100;
eta=25;
sigma=.5;
S=[lambda 0 0 1 0 0];
```

‘True’ System Simulator

```
function [yout,ydotout] = odex(t0,tfinal,y0,u,w,tol,trace)

global A B D K kls M n Ts

Knl=K+[0 0 0; 0 kls*round((1+sign(y0(2)))/4) 0; 0 0 0];
Anl=[zeros(n/2) eye(n/2);
     -(inv(M)*Knl) -(inv(M)*D)];

%ODEX is a customisation of the ode45 function
%
%ODE45     Solve differential equations, higher order method.
%     ODE45 integrates a system of ordinary differential equations
using
%     4th and 5th order Runge-Kutta formulas.
%     [T,Y] = ODE45('yprime', T0, Tfinal, Y0) integrates the system
of
%     ordinary differential equations described by the M-file
YPRIME.M,
%     over the interval T0 to Tfinal, with initial conditions Y0.
%     [T, Y] = ODE45(F, T0, Tfinal, Y0, TOL, 1) uses tolerance TOL
%     and displays status while the integration proceeds.
%
%     INPUT:
%     t0     - Initial value of t.
%     tfinal- Final value of t.
%     y0     - Initial value column-vector.
%       u     - input vector during integration interval
%       w     - system noise vector during integration interval
%     tol   - The desired accuracy. (Default: tol = 1.e-6).
%     trace - If nonzero, each step is printed. (Default: trace = 0).
%
%     OUTPUT:
%       yout    - system state vector at the end of interval
%       ydotout - derivative of the system state vector at the end
of interval
%
%     The result can be displayed by: plot(tout, yout).
%
%     See also ODE23, ODEDEMO.

%     C.B. Moler, 3-25-87, 8-26-91, 9-08-92.
%     Copyright (c) 1984-93 by The MathWorks, Inc.
%       J. Blansjaar, 4-11-1997

% The Fehlberg coefficients:
alpha = [1/4  3/8  12/13  1  1/2]';
beta  = [ [    1      0      0     0      0    0]/4
          [    3      9      0     0      0    0]/32
          [ 1932  -7200   7296     0      0    0]/2197
          [ 8341 -32832  29440  -845      0    0]/4104
          [-6080  41040 -28352  9295  -5643    0]/20520 ]';
gamma = [ [902880  0  3953664  3855735  -1371249  277020]/7618050
          [ -2090  0    22528    21970    -15048  -27360]/752400 ]';
pow = 1/5;
if nargin < 5, tol = 1.e-6; end
if nargin < 6, trace = 0; end

% Initialization
t = t0;
hmax = (tfinal - t)/16;
h = hmax/8;
y = y0(:);
f = zeros(length(y),6);
chunk = 128;
tout = zeros(chunk,1);
yout = zeros(chunk,length(y));
k = 1;
tout(k) = t;
yout(k,:) = y.';

if trace
   clc, t, h, y
end

% The main loop

while (t < tfinal) & (t + h > t)
   if t + h > tfinal, h = tfinal - t; end
   % Compute the slopes
   temp = Anl*y+B*u+w;
   f(:,1) = temp(:);
   for j = 1:5
      temp = Anl*(y+h*f*beta(:,j))+B*u+w;
      f(:,j+1) = temp(:);
   end

   % Estimate the error and the acceptable error
   delta = norm(h*f*gamma(:,2),'inf');
   tau = tol*max(norm(y,'inf'),1.0);

   % Update the solution only if the error is acceptable
   if delta <= tau
      t = t + h;
      y = y + h*f*gamma(:,1);
      k = k+1;
      if k > length(tout)
         tout = [tout; zeros(chunk,1)];
         yout = [yout; zeros(chunk,length(y))];
      end
      tout(k) = t;
      yout(k,:) = y.';
   end
   if trace
      home, t, h, y
   end

   % Update the step size
   if delta ~= 0.0
      h = min(hmax, 0.8*h*(tau/delta)^pow);
   end
end

if (t < tfinal)
   disp('Singularity likely.')
   t
end

ydotout=mean(f')';
yout=yout(k,:)';
```

Algebraic Reconstructor

```
%--------------------------------------
% algebra.m
%--------------------------------------

function [new_xhat,new_xhatdot,Tc]=algebra(xhat,xhatdot,y,u)

global A B K M Ts
```

```
tb=cputime;

%---------------------------------------
% algebraic reconstruction method
%---------------------------------------

new_xhat(1:2,1)=y(1:2);
new_xhat(4:5,1)=(y(1:2)-xhat(1:2))/Ts+((1/2)*xhatdot(4:5)*Ts);
new_xhatdot(1:2,1)=xhat(4:5);
new_xhatdot(4:5,1)=y(3:4);

A4=[A(4,1:2) A(4,4:5)];
A5=[A(5,1:2) A(5,4:5)];
x1245=[new_xhat(1:2); new_xhat(4:5)];

C1=(A(4,3)/A(4,6))-(A(5,3)/A(5,6));
C2=(new_xhatdot(4)-B(4,:)*u-A4*x1245)/A(4,6);
C3=(new_xhatdot(5)-B(5,:)*u-A5*x1245)/A(5,6);

new_xhat(3)=(C2-C3)/C1;

C4=(A(4,6)/A(4,3))-(A(5,6)/A(5,3));
C5=(new_xhatdot(4)-B(4,:)*u-A4*x1245)/A(4,3);
C6=(new_xhatdot(5)-B(5,:)*u-A5*x1245)/A(5,3);

new_xhat(6)=(C5-C6)/C4;

new_xhatdot(3)=new_xhat(6);
new_xhatdot(6)=A(6,:)*new_xhat+B(6,:)*u;

Tc=cputime-tb;
```

Kalman Filter

```
%---------------------------------------
% kalman.m
%---------------------------------------

function [new_xhat,Tc]=observe(xhat,u,y)

global A B C Ko kls Ts T11 T22

tb=cputime;

%---------------------------------------
% optimal kalman filter reconstruction method
% 4th order Runge-Kutta integration
%---------------------------------------

yhat=T11*xhat+T22*(A*xhat+B*u);
```

```
xi1=A*xhat+B*u+Ko*(y-yhat);

yhat=T11*(xhat+(Ts/2)*xi1)+T22*(A*(xhat+(Ts/2)*xi1)+B*u);
xi2=A*(xhat+(Ts/2)*xi1)+B*u+Ko*(y-yhat);

yhat=T11*(xhat+(Ts/2)*xi2)+T22*(A*(xhat+(Ts/2)*xi2)+B*u);
xi3=A*(xhat+(Ts/2)*xi2)+B*u+Ko*(y-yhat);

yhat=T11*(xhat+(Ts/2)*xi3)+T22*(A*(xhat+(Ts/2)*xi3)+B*u);
xi4=A*(xhat+(Ts/2)*xi3)+B*u+Ko*(y-yhat);

xhat=xhat+(Ts/6)*(xi1+2*xi2+2*xi3+xi4);

new_xhat=xhat;

Tc=cputime-tb;
```

Sliding Mode Observer

```
%---------------------------------------
% smo.m
%---------------------------------------

function [new_xhat,Tc]=smo(xhat,u,y)

global A B fi Ko Ksmo T11 T22 Ts

tb=cputime;

%---------------------------------------
% sliding mode observer reconstruction method
% 4th order Runge-Kutta integration
%---------------------------------------

yhat=T11*xhat+T22*(A*xhat+B*u);
xi1=A*xhat+B*u+Ko*(y-yhat)+Ksmo*sat(y-yhat,fi);

yhat=T11*(xhat+(Ts/2)*xi1)+T22*(A*(xhat+(Ts/2)*xi1)+B*u);
xi2=A*(xhat+(Ts/2)*xi1)+B*u+Ko*(y-yhat)+Ksmo*sat(y-yhat,fi);

yhat=T11*(xhat+(Ts/2)*xi2)+T22*(A*(xhat+(Ts/2)*xi2)+B*u);
xi3=A*(xhat+(Ts/2)*xi2)+B*u+Ko*(y-yhat)+Ksmo*sat(y-yhat,fi);

yhat=T11*(xhat+(Ts/2)*xi3)+T22*(A*(xhat+(Ts/2)*xi3)+B*u);
xi4=A*(xhat+(Ts/2)*xi3)+B*u+Ko*(y-yhat)+Ksmo*sat(y-yhat,fi);

new_xhat=xhat+(Ts/6)*(xi1+2*xi2+2*xi3+xi4);

Tc=cputime-tb;
```

Excitation

```
%-------------------------------------
% excite.m
%-------------------------------------

function u_excitation=excite(k)

global mere we Ts ts

u_excitation=mere*(we^2)*cos(we*(k-1)*Ts);
```

Control Force

```
%-------------------------------------
% control.m
%-------------------------------------

function [u_control,ex]=control(xhat,k)

global A B eta S sigma

[x_d,xd_d]=desitrac(k);
ex=x_d-xhat;
s=S*ex;
u_control=inv(S*B(:,1))*(eta*sat(s,sigma)+S*(A*ex+xd_d-A*x_d));
u_control=sign(u_control)*min(abs(u_control),20);
```

Desired Trajectory

```
%-----------------------------
% desitrac.m
%-----------------------------

function [x_desired,xd_desired]=desitrac(k)

global F we n Ts ts

q_d=F(1,1:3)';
qd_d=zeros(3,1);
qdd_d=zeros(3,1);
for i=1:10
  w=i*we;
  t=ts+(k-1)*Ts;
  q_d=q_d+cos(w*t)*(F(i+1,1:3)')...
         +sin(w*t)*(F(i+11,1:3)');
  qd_d=qd_d-w*sin(w*t)*(F(i+1,1:3)')...
           +w*cos(w*t)*(F(i+11,1:3)');
  qdd_d=qdd_d-(w^2)*cos(w*t)*(F(i+1,1:3)')...
             -(w^2)*sin(w*t)*(F(i+11,1:3)');
```

```
end
x_desired=[q_d; qd_d];
xd_desired=[qd_d; qdd_d];
```

Saturation Function

```
%----------------------------------
% sat.m
%----------------------------------

function factor=sat(a,b)

for i=1:size(a,1)
 if abs(a(i)./b(i))>=1
  factor(i,1)=sign(a(i)/b(i));
 else
  factor(i,1)=a(i)./b(i);
 end
end
```

Relative Euclidic Norm

```
%--------------------------------
% ren.m
%--------------------------------

function relative_euclidic_norm=ren(x,y)

for i=1:size(x,1)
  relative_euclidic_norm(i,1)=(norm(x(i,:)',2))/(norm(y(i,:)',2));
end
```