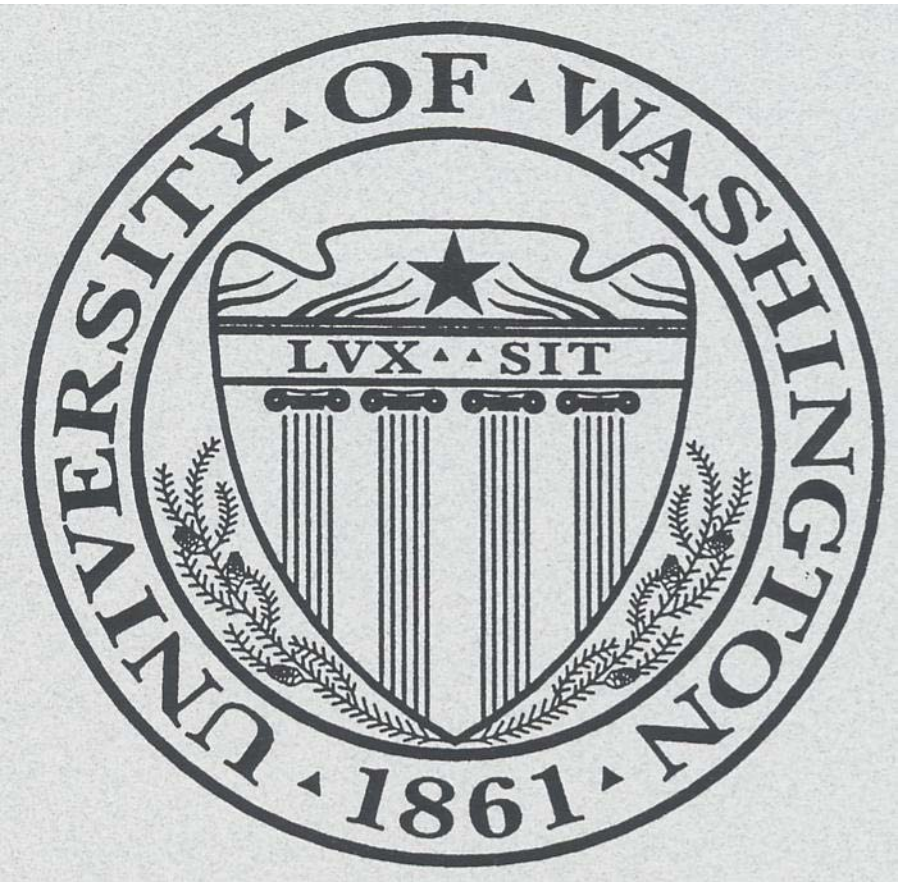
UNIVERSITY OF WASHINGTON
Department of Aeronautics and Astronautics

Parameter Identification and Adaptive Fault Detection of the Flexishaft System



June 9, 2004

Carlos Gonzalez
Daniel Klein
Christopher Lum

Table of Contents

Introduction

The goal of this project was to apply theories and ideas of adaptive control and estimation to an actual system. For this system we chose the Flexishaft experiment, which consists of two DC motors connected by a flexible shaft. Similar systems are used in manufacturing and various mechanical systems.

The Flexishaft is non-linear, as are all real systems. However, in the frequency range of interest, assumptions can be made which allow for the creation of an accurate linear model. This model serves as the basis for our estimation and adaptive control problem. We investigate several aspects of estimation and control including parameter identification, state estimation, and adaptive fault detection.

For parameter identification, we determine the input/output relationship of this system using a real time recursive least squares algorithm. We independently verify these results using the frequency response from a dynamic signal analyzer. The mathematical model, identified model from recursive least squares, and frequency response model from the DSA are then analyzed.

State estimation is performed using a discrete time Kalman filter algorithm. To make the design feasible to implement on hardware, we use the infinite time horizon Kalman gain. We compare results from estimating the state of a simulated plant to the state estimates of the actual hardware.

Finally, we use the real time parameter estimation in combination with adaptive control to achieve fault detection and compensation. This type of system has many applications in the field previously mentioned. By detecting faults automatically, a similar control system can act to preserve worker safety and equipment wellbeing in an industrial environment.

Acknowledgements

We would like to thank Professor Ly for allowing us to use the controls lab to perform this project.

Finally, many thanks to Professor Chizeck for his help and for canceling the final.

Experimental Apparatus

The experiment is based on the Flexishaft experiment in the controls lab. This system consists of two DC motors that are connected to each other through a flexible shaft. The general apparatus is shown below in Figure 1.

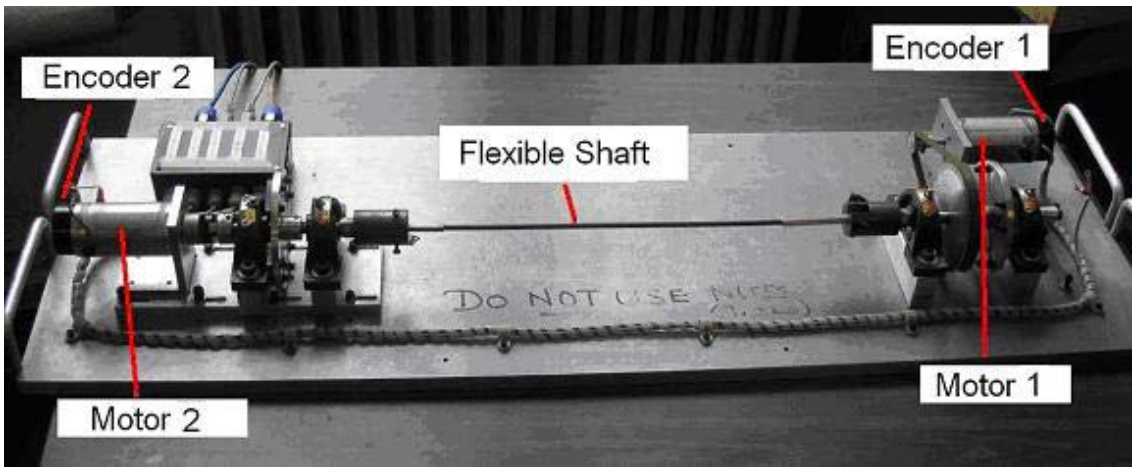


Figure 1: Flexishaft system

This can be considered in some situations as a square system since there are the same number of inputs and outputs. The inputs are voltages to two separate motors that can be controlled independently from one another. The outputs of the system are the angular position of the two motors. Since the rod connected the two motors is flexible, the angular positions of the two motors may be different and has dynamics associated with it.

There are three flexible shafts that can be used to connect the two motors. They are made out of steel, aluminum, and plastic. The steel is the stiffest and acts the most like a torsional spring. The plastic rod is the most flexible and has the most non-linear behavior.

The angles of the two motors are measured with two 1000 line/rev incremental encoders. The system inputs are limited to +/-10 volts (which is then amplified by 3.6 using an amplifier, which places saturation limits on the control voltage to +/-36 volts). There are no real constraints on the outputs.

In our application, we control motor 1 only and look at the response in the angular positions of the two motors. Note that motor 1 is not directly connected to the shaft and there is a 5:1 gear ratio between motor 1 and the angle θ_2.

The entire system is controlled real-time using xPC Target, which is a Simulink block set used for online control.

The two motors can be connected using several different shafts as shown below in Figure 2.

Figure 2: Different shafts for system

Notice that the plastic shaft is deformed and not straight. Therefore, it does not behave much like a torsional spring and causes problems with the system identification as shown in Parameter Identification section. Also notice that we have a breakable shaft that can be used to disconnect the two motors by removing the C-clamp. This is used to simulate a failure of the shaft.

The frequency response of the system is measured using a Hewlett Packard Dynamic Signal Analyzer as shown below in Figure 3.

Figure 3: Dynamic Signal Analyzer

System Modeling

The mechanical system is modeled using Newton's laws. The equations of motion governing the two motors are given by

$$\left(J_1 + n^2 J_{m1}\right)\ddot{\theta}_1 = nK_1 i_1 - \left(D_1 + n^2 D_{m1}\right)\dot{\theta}_1 - K_s\left(\theta_1 - \theta_2\right)$$
$$\left(J_2 + J_{m2}\right)\ddot{\theta}_2 = K_2 i_2 - \left(D_2 + n^2 D_{m2}\right)\dot{\theta}_2 + K_s\left(\theta_1 - \theta_2\right)$$

Equation 1

As can be seen, these are second order differential equations. We add two more states since we are not able to directly control the current and instead, we can only control the voltage applied to the amplifiers. We can model the system as a simple RLC circuit and we therefore have the following differential equations for the electrical model of the system

$$L_1\dot{i}_1 = -\left(R_{A1} + R_1\right)i_1 - K_1 n\dot{\theta}_1 + K_{A1}u_1$$
$$L_2\dot{i}_2 = -\left(R_{A2} + R_2\right)i_2 - K_2\dot{\theta}_2 + K_{A2}u_2$$

Equation 2

As can be seen, this is a six state system; we choose the state vector to be

$$x = \begin{pmatrix} i_1 & i_2 & \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 \end{pmatrix}^T$$

Equation 3

We then define the inputs and outputs to be

$$u = \begin{pmatrix} u_1 & u_2 \end{pmatrix}^T$$
$$y = \begin{pmatrix} \theta_1 & \theta_2 \end{pmatrix}^T$$

Equation 4

We can then write a linear state space representation of the system as

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$
$$y(t) = Cx(t)$$

Equation 5[1]

[1] For these matrices, see Appendix A

We then choose a sampling time of $T = 0.01$ seconds (see Parameter Identification for more information regarding sample time selection). With this, we can discretize the system using a zero order hold

$$A = e^{A_c T}$$
$$B = \left(\int_0^T e^{A_c \eta} d\eta \right) B_c$$

Equation 6

This yields the discrete state space equation of the system.

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

Equation 7

In the identification process, we are not able to identify the internal dynamics of the model (i.e. the state space representation). Instead, we are only able to identify the input/output relationship of the system. As stated previously, we are only interested in using the first control as an input, so we can calculate the corresponding two discrete transfer functions between the two outputs and the control u_1. These are both sixth order transfer functions of the general form.

$$G_i(z) = \frac{\theta_i(z)}{u_1(z)} = \frac{b_5 z^5 + b_4 z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0}{z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0}$$

Equation 8

Control Problem

Given the ability to perform tests on actual hardware, we felt that an interesting and useful application of adaptive control would be fault detection. It was our goal to identify through our parameter estimation a failure of the system consistent with the shaft breaking. If the shaft were to break, the system behavior would change markedly and we would be able to detect a shift in our parameter estimations at the time of the break. At this point, it would be desirable to stop the motors in order to prevent the broken shaft from continuing to spin. One can envision an application of such a fault detection system in various manufacturing environments.

In order to accomplish this, we required responsive identification of the parameters of the system as well as good convergence of these values. We would then use a very simple gain scheduling system that would command the motor to stop once certain criterion was met.

Assumptions

There are several assumptions that are made in the formulation of the mathematical model. First and foremost, the shaft is modeled as a simple torsional spring. This limits the shaft to only have a single mode. This translates into a single resonant peak for the frequency response of Equation 8. Furthermore, this assumption is probably the most accurate for the steel shaft since it is the stiffest and its first resonant peak is before the Nyquist frequency but all other resonant peaks are above the Nyquist frequency (see Figure 10 for more information).

Another assumption is that the friction of the system is modeled as purely viscous friction. This is a fairly bad assumption as the actual system has significant coulomb friction and stiction. The friction modeled in the linear system and a more accurate model of the actual friction of the system is shown below in Figure 4.

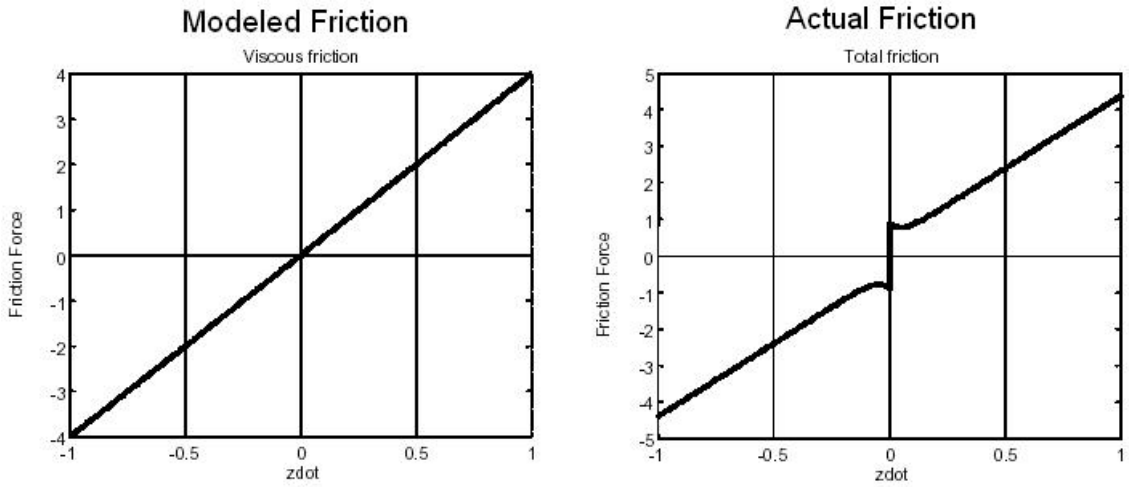


Figure 4: Linear and more accurate friction models[2]

As can be seen, this causes the most difference between the model and the actual system at low angular velocities. We see that if we add white noise of zero mean and bounded variance to the actual input signal, we can prevent the system from encountering in the stiction force. This method of dithering the signal improves the correlation between the actual system and the mathematical model. The main reason for modeling the friction as viscous friction is because it is a linear phenomenon and therefore, the system can be written in state space form.

In addition to assumptions made during the modeling process, there are assumptions we made during the identification process.

[2] Taken from Friction Modeling in Linear Track Cart Pendulum System, by Kalev Sepp. http://www.aa.washington.edu/courses/aa449/frictionmod99.pdf

The process of choosing parameters for our recursive least squares implementation was made significantly easier due to the fact we had a linear model describing the system. Given the assumption that the model was accurate for the operating range of our system, we knew that we could use a linear representation of a sixth order system in our recursive least squares formulation. Furthermore, the discretized equations of the model told us that the system had an input/output delay of one. We chose a recursive approach as opposed to a batch approach due to ease of computation and because our adaptive control goals were most suited to a recursive estimation implementation.

We also assume that the error in the system was equation error. This means we avoid implementing the extended recursive least squares, which makes the computational load less. It will be shown in the Parameter Identification section that this is a valid assumption and yields a very accurate identified model.

Our initial conditions and P_0 matrix were chosen to mimic having little knowledge of the system beforehand. Therefore, our initial condition was set to zero and our P_0 matrix was set to 100*I.

Finally, in order to ensure that we were satisfying the persistent excitation requirement for identification, we decided to use zero mean white noise on top of our control signal as input to the system.

Parameter Identification

The goal of this section is to determine numerical values for the coefficients of the transfer function, which governs the input/output relationship of the system. The numerical values of the coefficients and the structure of the discrete model are dependent on the sampling period used. We choose a sampling time of

$$T = 0.01 \text{ sec}$$

Equation 9

This value was chosen because from the continuous mathematical model, we know that the first resonant peak of the steel shaft is around 200 rad/s. Therefore, choosing this sampling time yields a Nyquist frequency of roughly 300 rad/s, so we can capture the first resonant peak. This places a restriction on the upper limit of the sampling time. Furthermore, we did not choose a faster sampling period to limit the amount of data that was produced and the ease computational cost. This places a lower bound on the sampling time.

We can also see the effect of different sampling times on the poles and zeros of the mathematical model system as shown below in Figure 5.

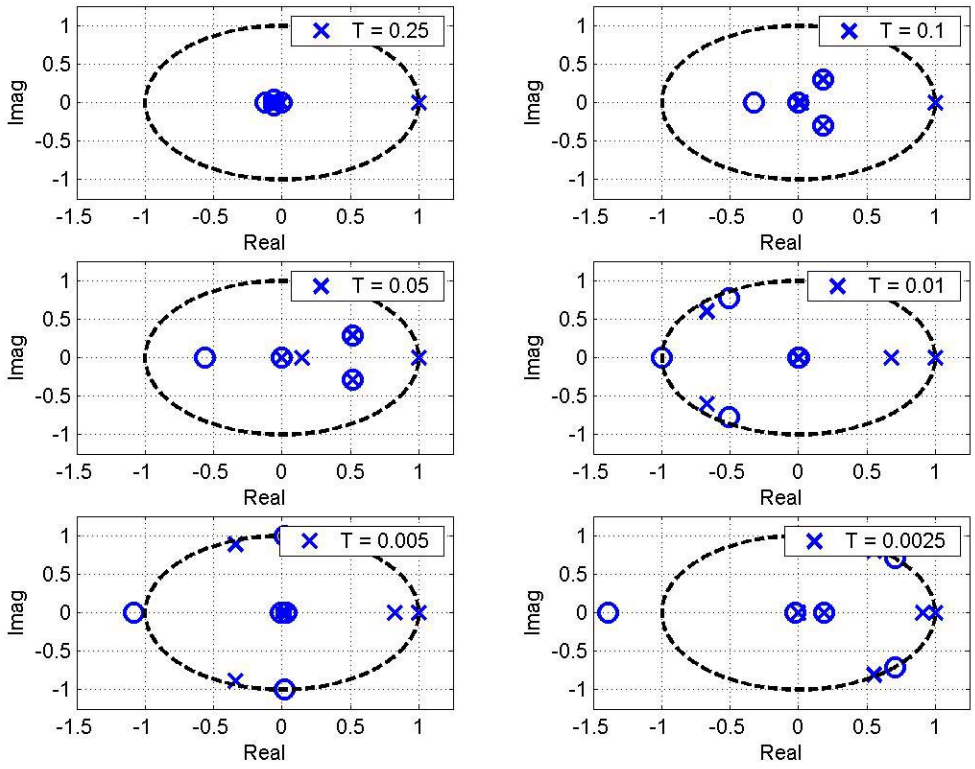


Figure 5: Pole zero map of $G_1(z)$ as a function of sampling time

The chosen sampling period of 0.01 seconds yields a minimum phase system. Sampling any faster would yield a non-minimum phase system. In general, as the sampling time increases, the poles and zeros of the system move away from the origin. Also notice that although the zeros leave the unit circle, the poles all remain stable no matter the sampling period.

With the sampling time chosen, we excite the system with white noise with zero mean and variance = 3. The input and outputs of the system are shown below in Figure 6.

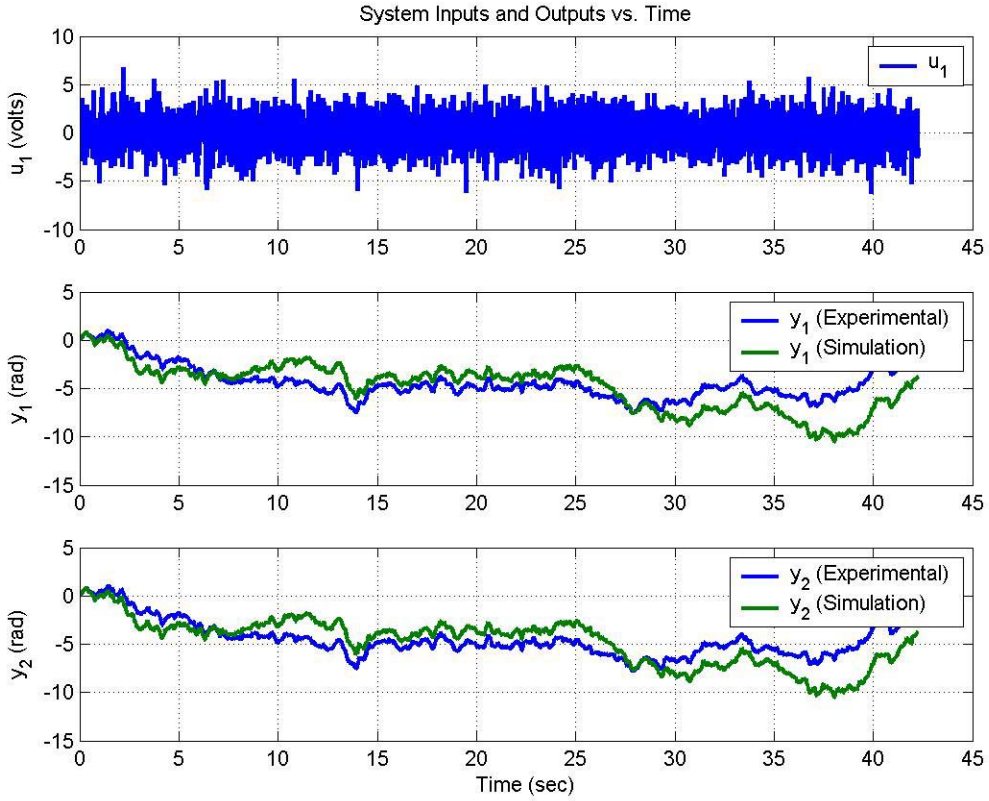


Figure 6: Input and output for system when u_1 is white noise, zero mean, variance = 3

As can be seen, the simulation model matches the actual experimental data fairly well. Also we see that the actual noise appears to have zero mean since the system outputs oscillate near zero in both situations. We use the on-line form of the recursive least square algorithm to estimate the parameters of the system in real time. The recursive least squares algorithm is shown below.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1))$$
$$K(t) = P(t-1)\varphi(t)(\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1}$$
$$P(t) = (I - K(t)\varphi^T(t))P(t-1)/\lambda$$

Equation 10

We initialize this algorithm with $P(0) = 100 * I$ and $\hat{\theta}(0) = zeros(12,1)$. Furthermore, we know that the parameters are not time varying, so we use λ=1. We can then first estimate the parameters for the $G_1(z)$ transfer function (between θ_1 and u_1). This yields the following estimates for the denominator polynomials.

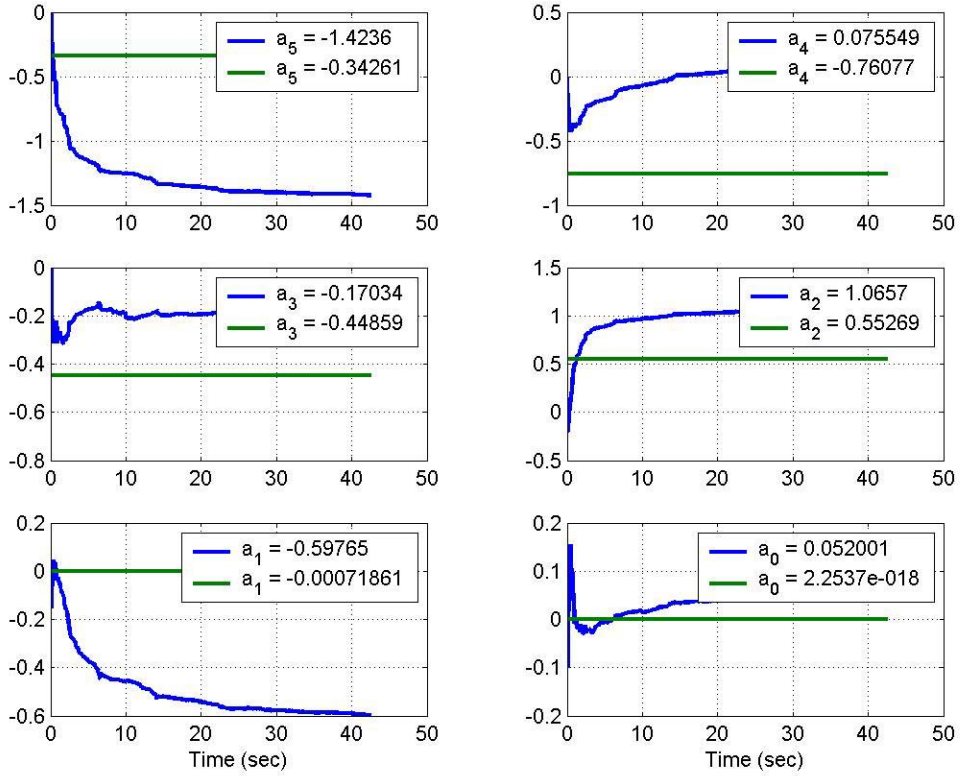


Figure 7: Denominator coefficients for estimating $G_1(z)$

Shown in blue are the real time estimates of the actual hardware system. Shown in green are the coefficients of the transfer function from the mathematical model (see Appendix A for more information). As can be seen, the recursive least square model converges to constant parameter estimates. Notice that the parameter estimates do not match the mathematical model very closely. This probably means that the mathematical model is somewhat inaccurate. Recall that there were several major assumptions when modeling the system, so this is not a surprise. The numerator coefficients are shown below in Figure 8.

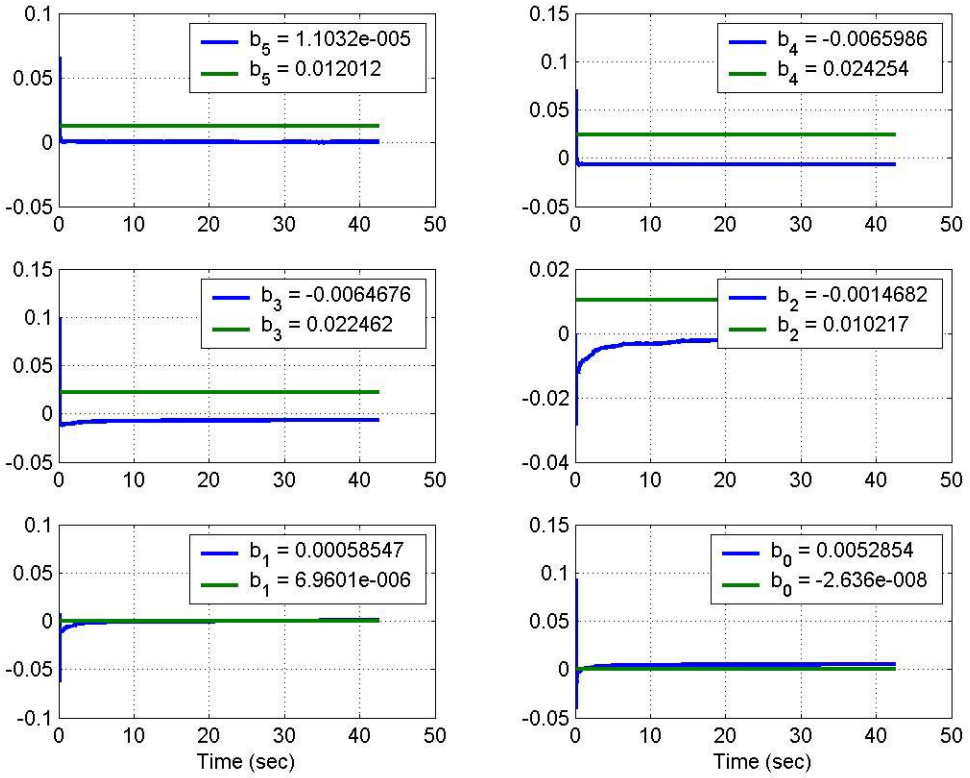


Figure 8: Numerator coefficients for estimating $G_1(z)$

The numerator coefficients converge much faster to their final values. Furthermore, we see that these match the mathematical model much better. The final identified transfer function that we arrive at is given by

$$G_1(z) = \frac{\theta_1(z)}{u_1(z)} = \frac{1.103\times10^{-5}z^5 - 0.006599z^4 - 0.006468z^3 - 0.001468z^2 + 0.0005855z + 0.005285}{z^6 - 1.424z^5 + 0.07555z^4 - 0.1703z^3 + 1.066z^2 - 0.5976z + 0.052}$$

Equation 11

To check the correlation between the identified model and the mathematical model, we can pole zero map the two transfer functions. This yields Figure 9.

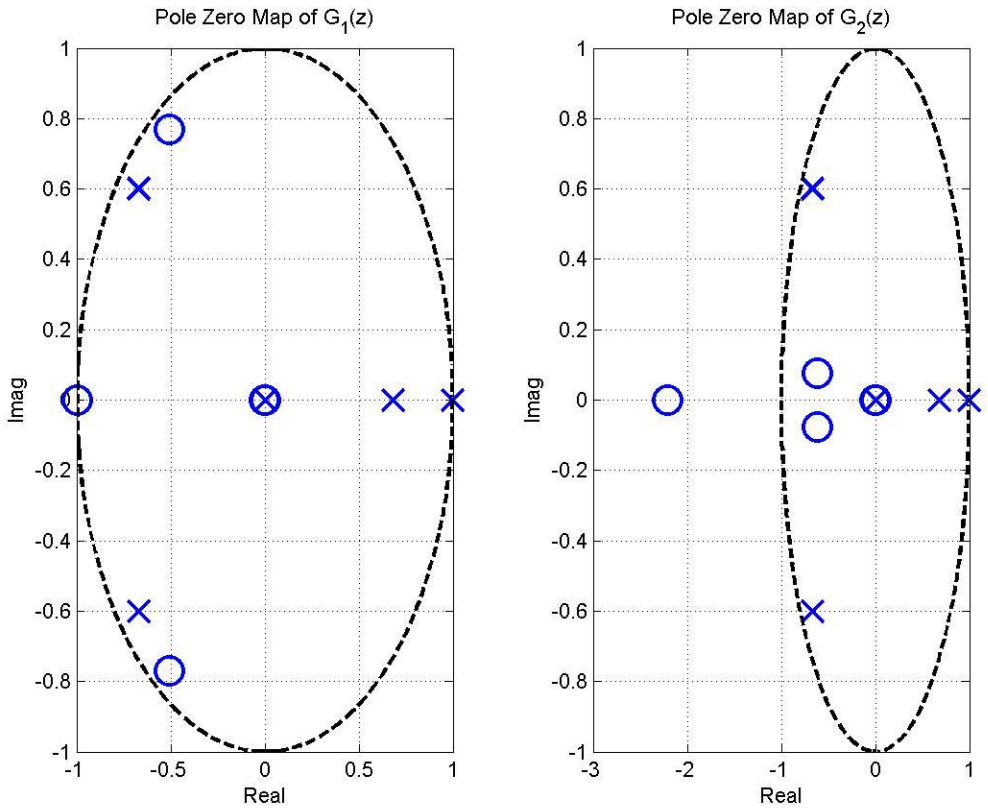


Figure 9: Pole zero map of identified and mathematical model transfer function

As can be seen, the identified system and the mathematical model do not match exactly. However, we see that the general location of the poles and the zeros are the same for both systems. Also note that the poorly damped poles and zeros (the ones near the unit circle) match fairly well. This is because the effects of these poles and zeros are the most obvious in the output since they are the dominant poles and zeros. Also notice that with the non-colocated system has a non-minimum phase zero whereas the collocated system is minimum phase.

We need to now independently verify these parameter estimates. In order to do this, we use the Dynamic Signal Analyzer and identify the frequency response of the actual system. We can then evaluate the frequency response of the identified transfer function and compare the two. Although the transfer functions are not the exact same (as shown in Figure 7 and Figure 8), it is possible that they have similar input/output relationship at certain frequencies. The frequency response of the actual system, the identified model, and the mathematical model are shown below in Figure 10.

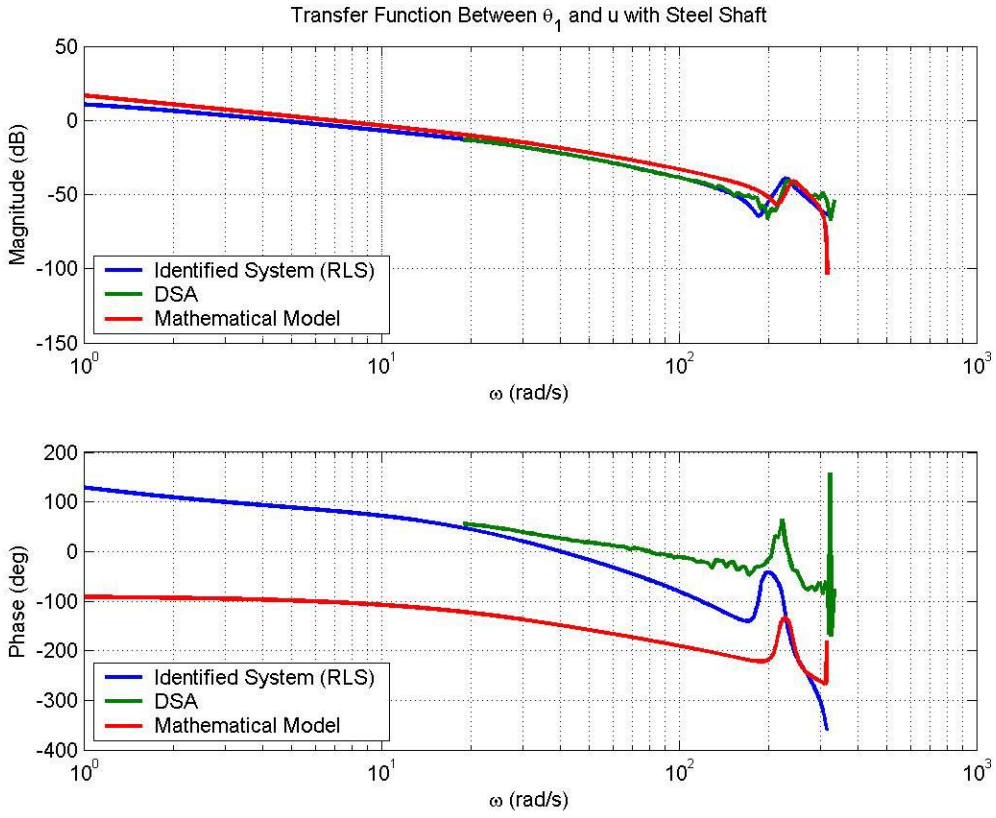


Figure 10: Frequency response of system with steel shaft, variance = 3, $y=\theta_1$

The identified model using the recursive least squares algorithm is shown in red, the actual frequency response of the system using the DSA is shown in green, and the mathematical model is shown in blue. As can be seen, the magnitudes of all three match very well. This shows that although the actual transfer functions are not exactly the same, the input output relation ship is very similar for the frequencies up to the Nyquist frequency.

Notice however that the phase is not exactly the same but the general trend is there. Also, the resonant frequency of this system is accurately identified. Because this shaft is so stiff, the response between the control and the second output is nearly the same as the first input.

As predicted the steel shaft acts linearly like a torsional spring. The plastic shaft, due to the warping and flexibility, exhibits the most non-linear behavior and therefore, the identified model and the DSA response are significantly different. This data is shown in Appendix B. Notice however that in both cases, the identified model does correctly identify the resonant frequency of the system.

State Estimation

As stated previously, although we can identify the parameters of the transfer function, the recursive least squares does not allow us to estimate the physical state of the plant. The state estimation problem seeks to find the state of a system given a history of past inputs and outputs. Clearly, state estimation implies that the system model must have some state and therefore must be represented in state space form.

Methodology and Implementation

We implemented a discrete time Kalman filter for state estimation. For linear systems with Gaussian process and observation noise, the Kalman gives a minimum variance estimate of the state recursively. However, the physical Flexishaft system is non-linear (as are all physical systems) and noises likely not Gaussian. However, we were still able to get accurate state estimations with this filter.

The discretized state space representation of the system is given in Equation 7. The basic idea of the state observer is to maintain a model of the system in parallel to the actual system. The main difference between the observer model system and the actual system is that there as an added input to observer that drives the observer state to the actual state of the system. The equations for the observer are

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + L(t)[y(t) - C\hat{x}(t)]$$
$$\hat{y}(t) = C\hat{x}(t)$$

Equation 12

The Q matrix represents how noise enters the system. Assuming that the noise enters into the process through the control input, we chose Q to be BQ_wB^T with $Q_w = I$ since we assume that both inputs have the same process noise. The R matrix is the sensor covariance matrix. In this case, since we have no real information regarding the variance of the encoders, we choose R equal to the identity matrix (effectively saying that both sensors are equally noisy).

$$Q = BQ_wB^T$$
$$R = I$$

Equation 13

The optimal Kalman gain, L, is time varying and is obtained by solving the discrete Ricatti Equation forward in time. However, if we use the steady state value, we obtain the time invariant Kalman gain that is given by

$$L^T = \left[R + CMC^T \right]^{-1} CM$$

Equation 14

Where M is the solution to the discrete Algebraic Ricatti Equation

$$M = A^T \left\{ M - MC^T \left[CMC^T + R \right]^{-1} CM \right\} A^T + Q$$

Equation 15

L could be obtained by solving the discrete Ricatti Equation (time varying) forward in time and then using the steady state value of L as the time invariant L matrix. However, to make computation easier, we simply use

$$L = dlqr(A^T, C^T, Q, R)^T$$

Equation 16

The last step was to create a Matlab Simulink® model of the system and observer. We actually created two similar Simulink models, one for the mathematical model plant to be run offline and another for the real plant to be run online. Following the Kalman filter equations, we produced the models shown in Figure 11 through Equation 12.

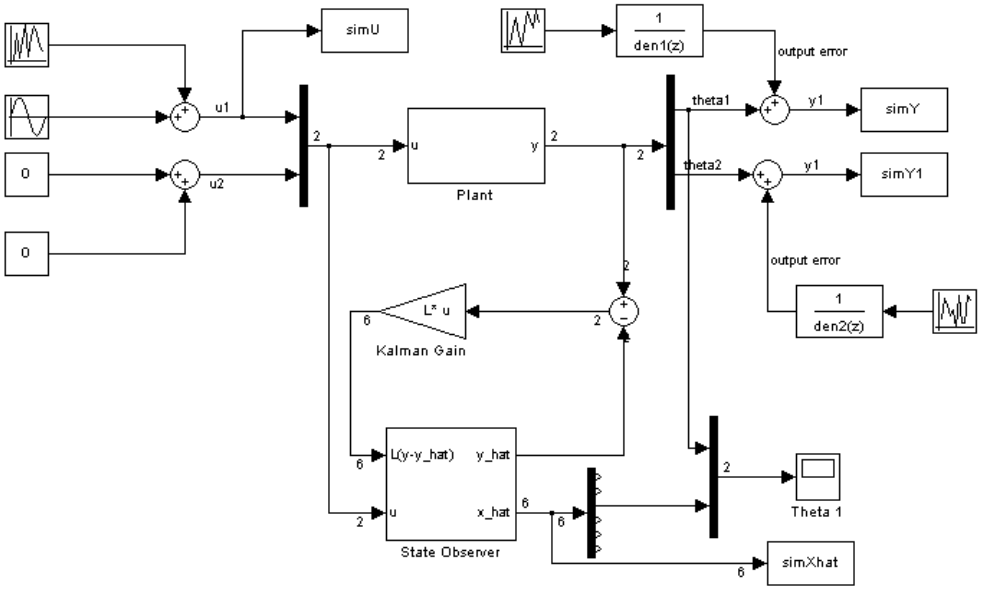


Figure 11: Simulink model used for simulated state estimation from a simulated plant.

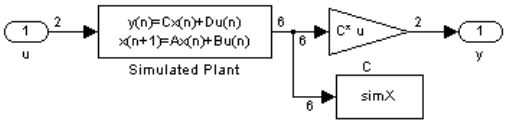


Figure 12. Simulink model of simulated plant.

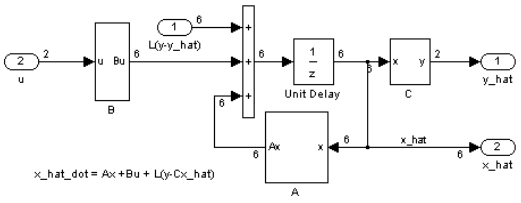


Figure 13. Simulink model of the state observer.

State Estimation of the Mathematical Model Plant

For the mathematical model plant, the estimated states quickly and smoothly approach the actual values. Even when the initial conditions of the observer were wildly different from those of the plant, only a half-second was needed for the estimated values to converge. See Figure 14 for the estimate of θ_1 and corresponding system input. The estimate of θ_2 was nearly identical to that of θ_1 and was therefore omitted.

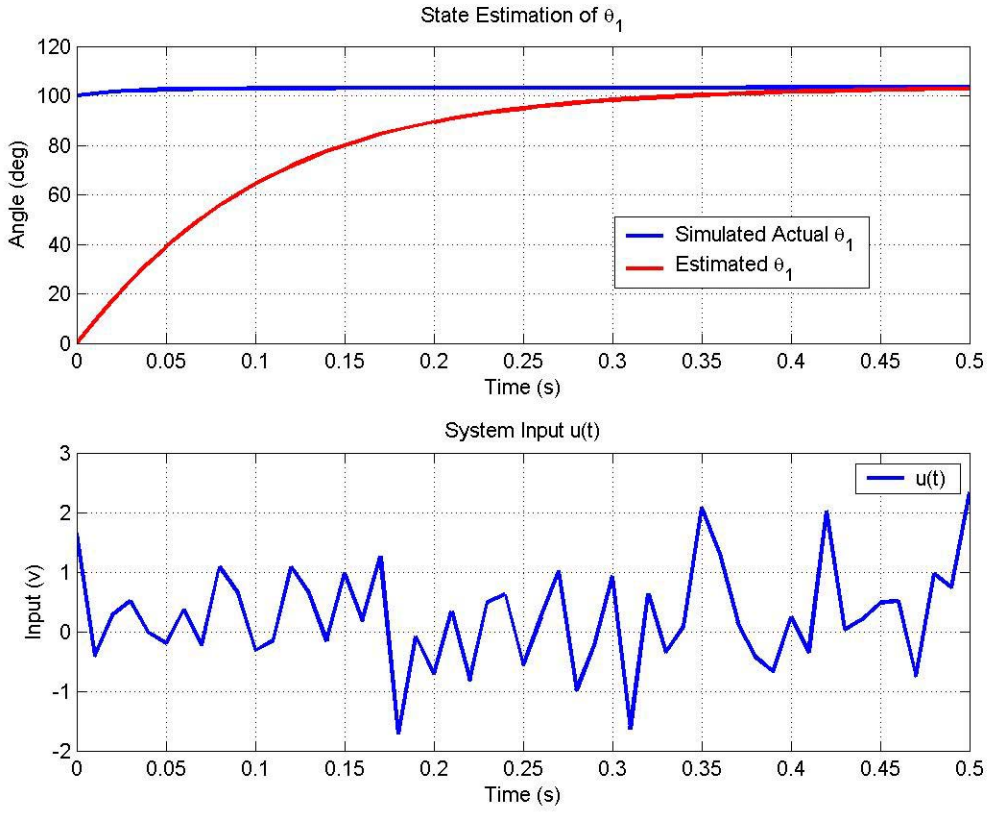


Figure 14. Estimation of θ_1 state and corresponding u(t) for the simulated plant.

State Estimation of the Actual Flexi-Shaft Plant

There was considerably more noise in the real plant than the simulated plant, yet the state estimates were still good. The noise in the plant was partially driven by the noise we intentionally added to the input signal to give good parameter identification (discussed in the next section).

Further, the A matrix in the Kalman observer was the mathematical model which we know to be different from the actual hardware. See Figure 15 for the estimates of θ_1 and the corresponding input signal u(t). The estimate of θ_2 looks identical to that of θ_1.

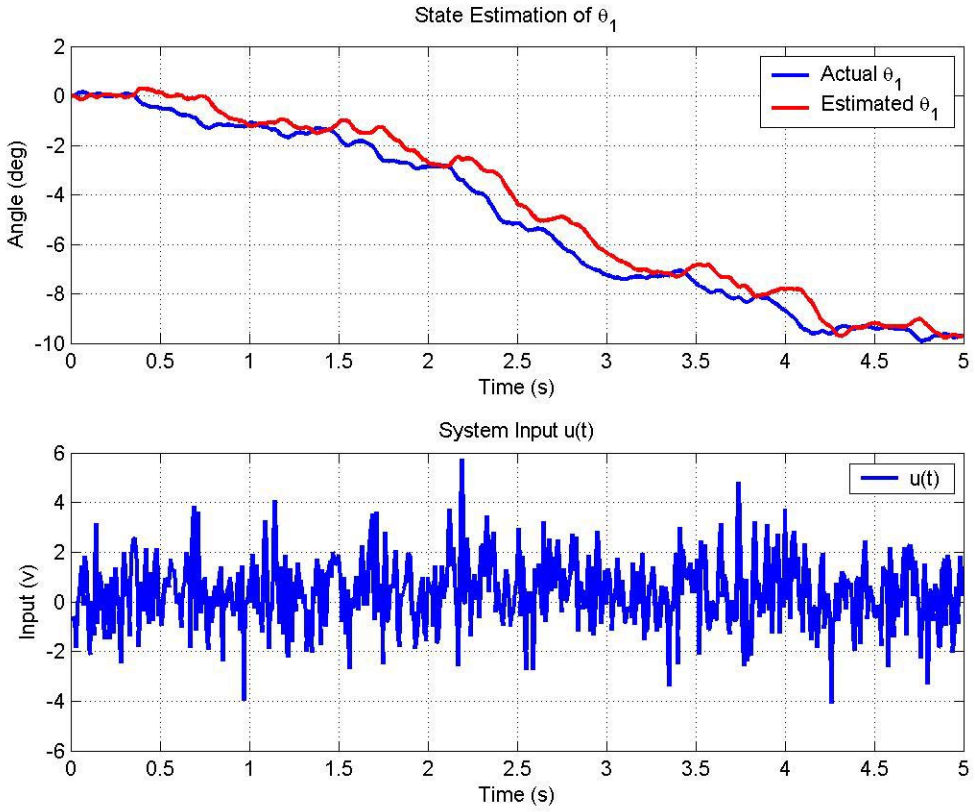


Figure 15. Estimation of θ_1 state and corresponding u(t) from the actual plant.

Adaptive Control

As shown in the Parameter Identification section, the parameter estimates for the system exhibited good convergence and settled to steady-state values. This means that any changes in the behavior of the system would cause a distinct shift in the parameter estimates. The goal is to detect changes characteristic to failure of the shaft and then shut the system off.

Given that the system parameters are now time varying, there was a need to choose a value for λ. A low forgetting factor would make our parameters estimates response faster but we run the risk that instead of identifying the actual system parameters, we would be identifying the noise. Through experiment, we found that the high value for λ, 0.99, gave us acceptable convergence.

Another method to deal with time varying parameters is to run multiple estimators in parallel and reset the P of each estimator, in sequence, to a large value. The advantage of this is that there is little risk of fitting noise, because λ is 1. However, we decided to not use this method.

Through experimentation, it was found that the a_2 parameter was particularly sensitive to changes in the system. A threshold value of 0.85 was crossed consistently upon failure of the shaft. This was implemented in Simulink as shown below in Figure 16.

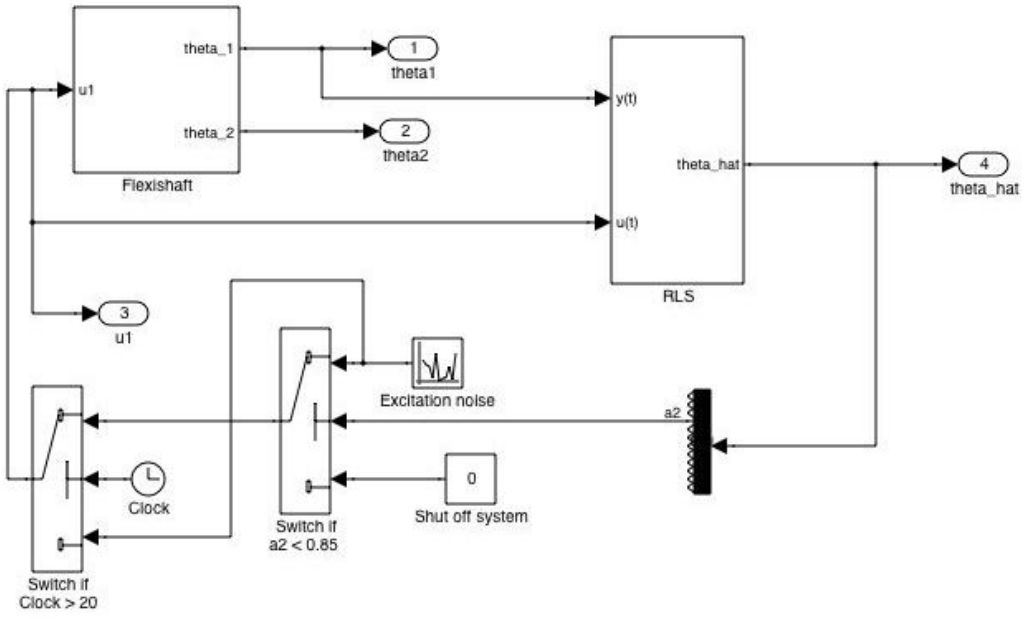


Figure 16: Real time system for parameter estimation and fault detection

The clock signal in the model is required because a_2 is initially zero, which is below the threshold value of 0.85.

It should be noted that a more robust failure-detection mechanism would detect a threshold rate of change for the parameter values as opposed to a threshold value. Another enhancement would be to use more than one parameter value as markers for a change in the system. However, we found that using a threshold based on a single parameter provided good response of the system to failure.

While it was not feasible to actually break the steel shaft while it was rotating, two separate lengths of metal were linked together with C-clamps as shown in Figure 2. To simulate a break, the clamp was loosened as quickly as possible to separate the shaft into two pieces.

Adaptive Control Results

The Simulink model given in the previous section was used to drive the shaft. At t=25 seconds, the clamps were loosened as described in the previous section. By t=30 seconds, the shaft had been separated into its two constituent pieces. The variation in the parameter estimates can be seen below in Figure 17.

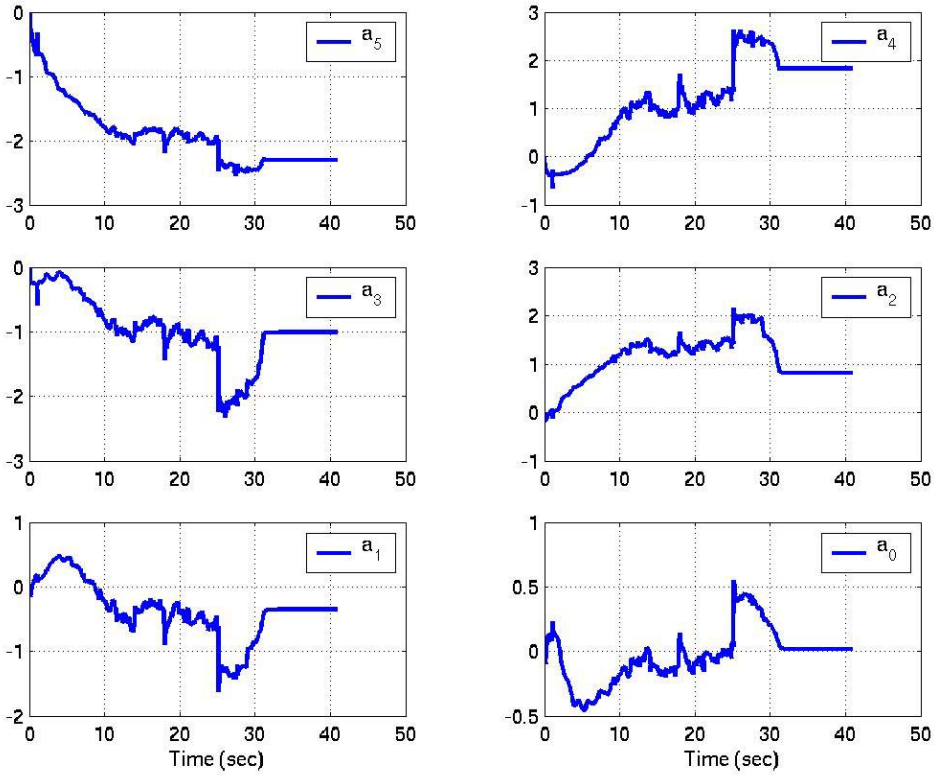


Figure 17: Denominator polynomial parameter estimates as shaft is being broken

It is clear that the estimates start changing as soon as the clamps on the shaft are manipulated. In our graph of a_2 estimates, we see the value jump up at t=25. However, it is the dip at t=30 which we were interested in detecting. Once we were done manipulating the clamps, our parameter estimates began to migrate to their new steady-state values. It is interesting to note that the failure of the shaft is not evident in the graph of θ_1 as shown below:

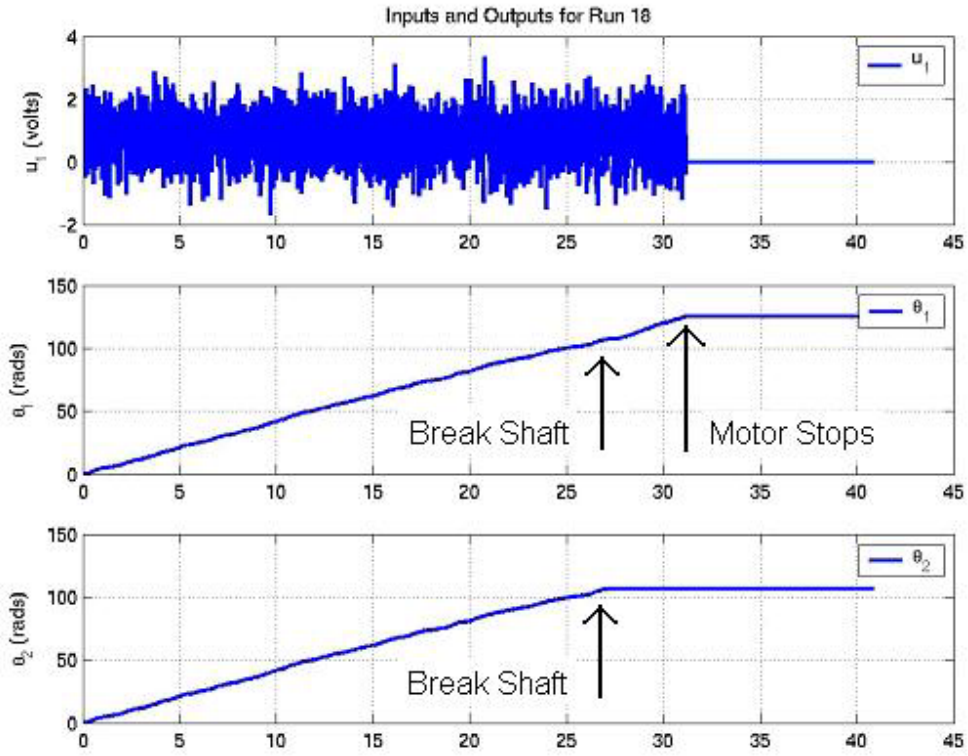


Figure 18: Inputs and outputs when breaking shaft

This further motivates the detection of failure by examining the parameter estimates. While it is clear from the θ_2 output that a failure has occurred, if the system only provided θ_1 output there would be no way of telling that failure had occurred. The θ_1 values continue varying linearly until the system shuts off at approximately 31 seconds. At this time, the a_2 estimate had crossed the threshold value and the motor was commanded to stop. A video of the system shutting itself off after failure can be seen at http://vger.aa.washington.edu/shaftbreak.mov

Conclusions

This project allowed us to gain valuable experience in the area of adaptive control and parameter identification. We managed to complete all three of our goals (parameter identification, state estimation, and adaptive fault detection) successfully in the last day before it was due (no, just kidding).

The system behaved as expected throughout our experiments. The algorithms worked well in general and gave us meaningful results. These results lead us to believe that the various assumptions and approximations we made were appropriate.

In parameter estimation, we saw good parameter convergence, although not always to the expected values. Again, we attribute this to non-linear behavior inherent in the system.

Finally, our fault detection system worked well. It was gratifying to see the system stop itself shortly after failure. This brought together several aspects we covered in the course and it was very interesting to see them come together.

Throughout the project, we had to overcome various limitations of the software drivers which controlled the hardware. Namely, xPC Target does support MATLAB functions or standard S-functions. Unfortunately, many of the algorithms we had previously prepared were written as either MATLAB functions or as S-functions. We were then forced to re-implement all of these algorithms using standard Simulink blocks. This task would have been easier if we had known how to use MATLAB C-MEX functions, which are supported by xPC Target. These functions would have allowed us to use C code to drive the hardware.

Appendix A

Matrices for linear state space representation (continuous time)

$$A_c = \begin{pmatrix} -\frac{R_{A1}+R_1}{L_1} & 0 & 0 & -\frac{nK_1}{L_1} & 0 & 0 \\ 0 & -\frac{R_{A2}+R_2}{L_2} & 0 & 0 & 0 & -\frac{K_2}{L_2} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{nK_1}{J_1+n^2J_{m1}} & 0 & -\frac{K_S}{J_1+n^2J_{m1}} & -\frac{\left(D_1+n^2D_{m1}\right)}{J_1+n^2J_{m1}} & \frac{K_S}{J_1+n^2J_{m1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{K_2}{J_2+J_{m2}} & \frac{K_S}{J_2+J_{m2}} & 0 & -\frac{K_S}{J_2+J_{m2}} & -\frac{\left(D_2+D_{m2}\right)}{J_2+J_{m2}} \end{pmatrix}$$

$$B_c = \begin{pmatrix} \frac{K_{A1}}{L_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_{A2}}{L_2} & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrices for linear state space representation (discrete time)

A =

-0.0037	-0.0102	-18.6908	-0.0522	18.6908	-0.1035
-0.0033	0.0019	0.9077	-0.0434	-0.9077	0.0117
0.0004	0.0002	0.5314	0.0047	0.4686	0.0035
0.0186	0.0474	98.2172	0.2681	-98.2172	0.4609
0.0005	0.0002	1.2240	0.0066	-0.2240	0.0021
0.0693	-0.0239	-135.5729	0.8650	135.5729	-0.2310

B =

1.1562	-0.0857
-0.0857	2.5490
0.0120	0.0012
1.8572	0.4278
0.0082	0.0032
2.5301	0.3952

Mathematical Model Transfer Functions (Steel Shaft)

$$G_1(z) = \frac{\theta_1(z)}{u_1(z)} = \frac{0.01201z^5 + 0.02425z^4 + 0.02246z^3 + 0.01022z^2 + 6.96\times10^{-6}z - 2.626\times10^{-8}}{z^6 - 0.3426z^5 - 0.7608z^4 - 0.4486z^3 + 0.5527z^2 - 0.0007186z}$$

$$G_2(z) = \frac{\theta_2(z)}{u_1(z)} = \frac{0.00823z^5 + 0.02831z^4 + 0.02548z^3 + 0.006938z^2 - 8.516\times10^{-6}z - 2.285\times10^{-10}}{z^6 - 0.3426z^5 - 0.7608z^4 - 0.4486z^3 + 0.5527z^2 - 0.0007186z}$$

Time invariant Kalman gain

$$L = \begin{pmatrix} -0.8329 & -0.1756 & 0.2470 & 3.9069 & 0.2503 & 3.4654 \\ -0.6796 & -0.1797 & 0.2469 & 3.1407 & 0.2469 & 4.1766 \end{pmatrix}^T$$

Appendix B

Aluminum Shaft

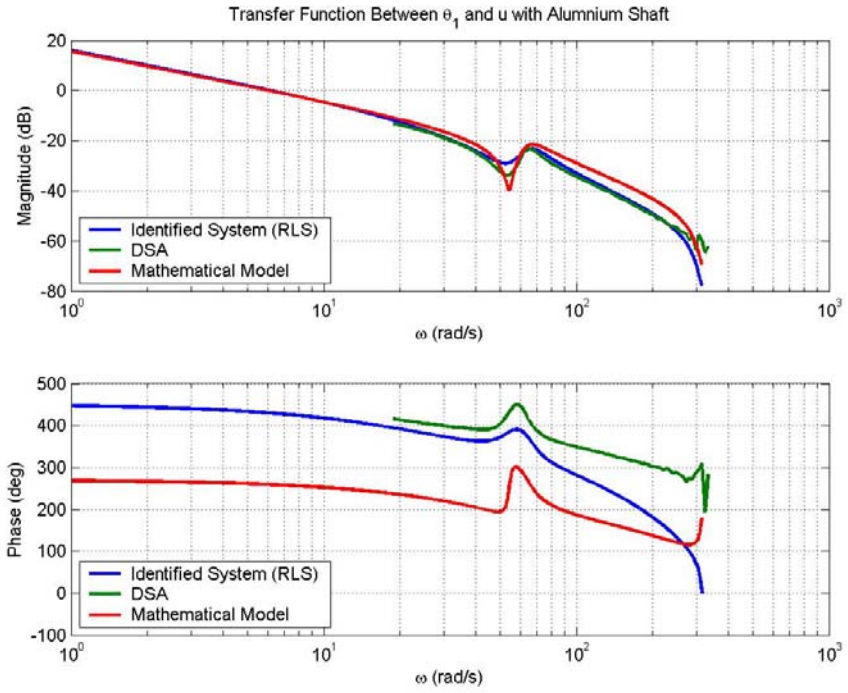


Figure 19: Frequency response of system with aluminum shaft, variance = 3, $y=\theta_1$

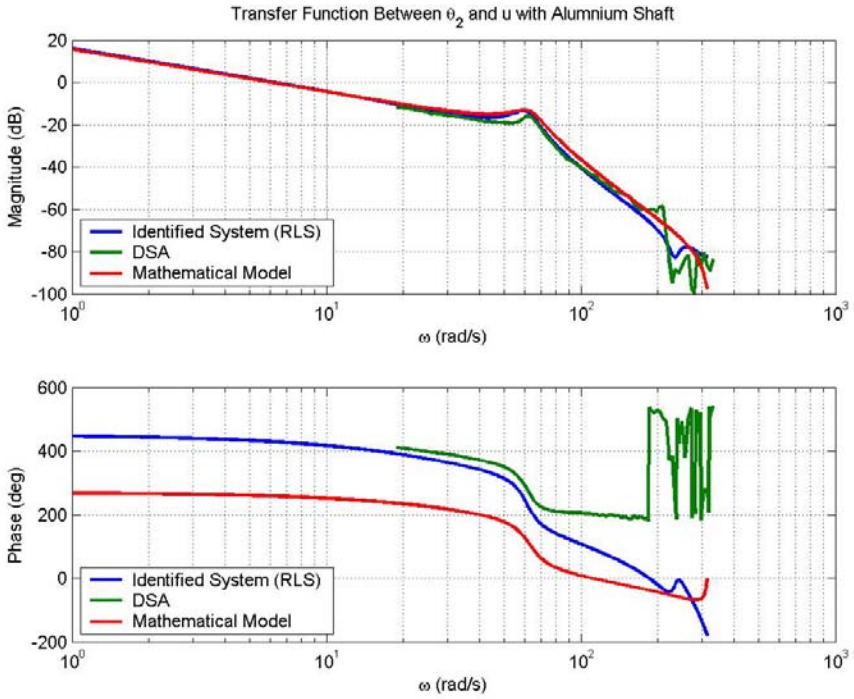


Figure 20: Frequency response of system with aluminum shaft, variance = 3, $y=\theta_2$

Plastic Shaft

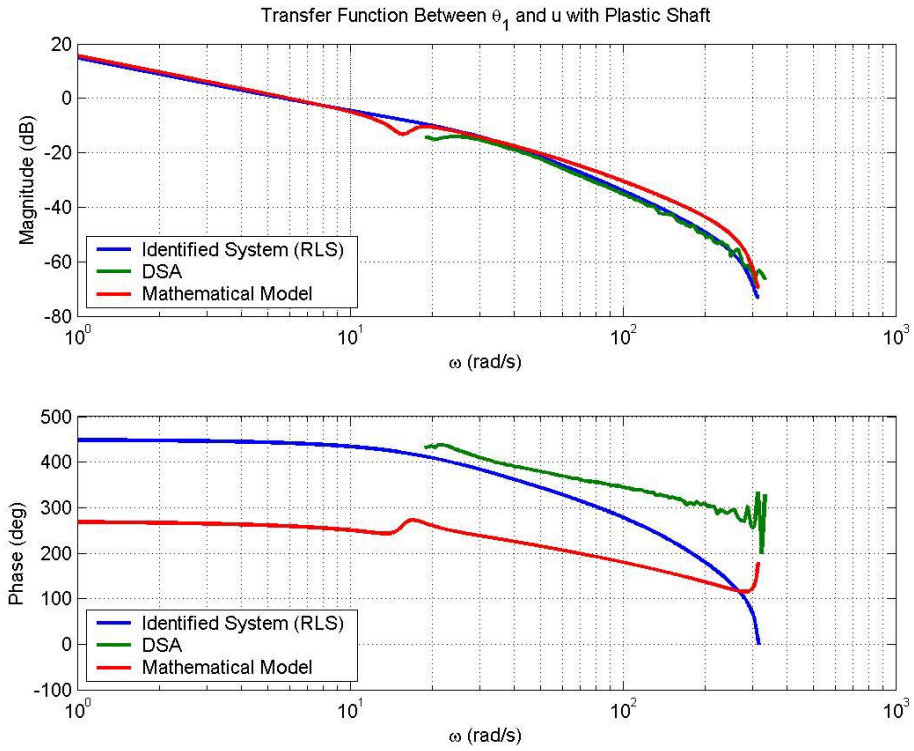


Figure 21: Frequency response of system with plastic shaft, variance = 3, $y=\theta_1$

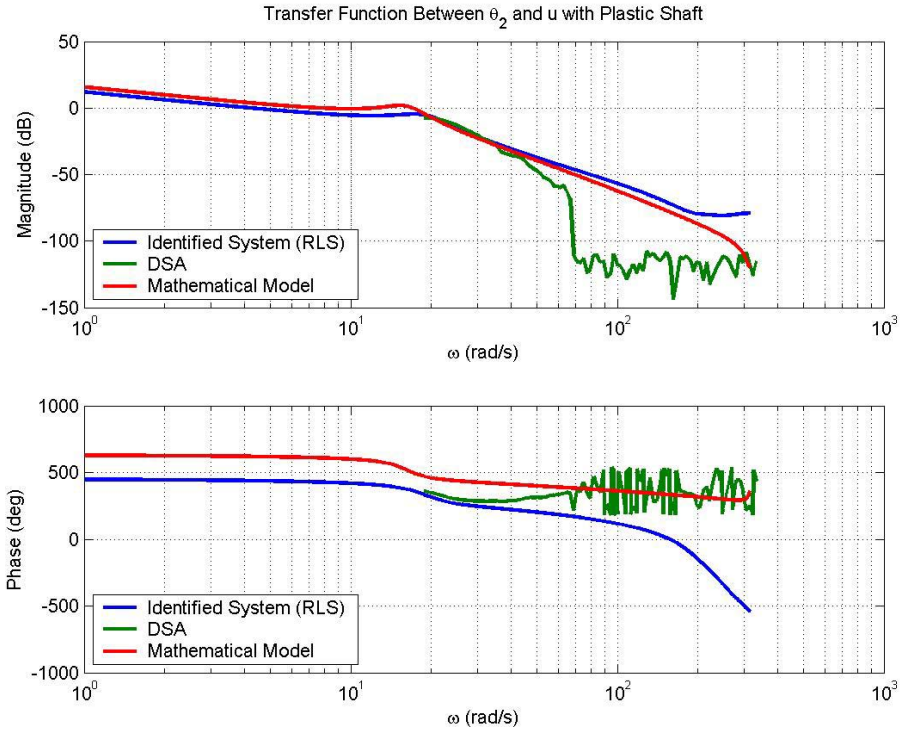


Figure 22: Frequency response of system with plastic shaft, variance = 3, $y=\theta_2$

Appendix C

Table 1: Symbols and variables used in this report

Symbol	Value/Units	Description
A		Discrete time A matrix
A_c		Continuous time A matrix
B		Discrete time B matrix
B_c		Continuous time B matrix
C		C matrix
D		D matrix
D_1	0.00004235	Viscous friction coefficient 1 (kg m^2)
D_2	0.00004235	Viscous friction coefficient 2 (kg m^2)
D_{m1}	0.000127	Motor 1 damping coefficient (Nm/(rad/s))
D_{m2}	0.000060	Motor 2 damping coefficient (Nm/(rad/s))
$G_1(z)$		Transfer function between θ_1 and u_1
$G_1(z)$		Transfer function between θ_2 and u_1
J_1	0.000125	Inertia load on motor 1 shaft (kg m^2)
J_2	0.0001	Inertia load on motor 2 shaft (kg m^2)
J_{m1}	0.0000269	Motor 1 inertia (kg m^2)
J_{m2}	0.00002438	Motor 2 inertia (kg m^2)
K_1	0.099	Motor constant 1 (V/(rad/s))
K_2	0.062	Motor constant 2 (V/(rad/s))
K_{A1}	3.6	Motor 1 amplifer (V/V)
K_{A2}	3.6	Motor 2 amplifer (V/V)
K_s	100 (Steel) 3 (Aluminum) 0.25 (Plastic)	Shaft Stiffness (Nm/rad)
L_1	0.000686	Armature inductance 1 (H)
L_2	0.0021	Armature inductance 2 (H)
n	5	Gear ratio
R_1	2.13	Armature resistance 1 (Ω)
R_2	1.2	Armature resistance 2 (Ω)
R_{A1}	0.2	Amplifier 1 resistant (Ω)
R_{A2}	0.2	Amplifier 2 resistant (Ω)
θ		Angular position of motors (rad)
u_1		Input to drive motor (volts)
x		State vector
y		Outputs of system