A TECHNICAL REPORT

On

EXTERNAL FORCE ESTIMATION OF IMPEDANCE-TYPE DRIVEN MECHANISM FOR SURGICAL ROBOT WITH KALMAN FILTER

Submitted by

SRIKUMAR MURALIDHARAN – 116950572

and

KUMAR SAMBHAV MAHIPAL – 116908108

For the course

ENPM667 – CONTROL OF ROBOTIC SYSTEMS

Course Instructor –

Dr. WASEEM MALIK

and

Dr. SESAN IWARERE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1:

Introduction

Surgical Robots have greatly contributed in the development of Minimally invasive surgery. Minimally invasive surgery involves tiny incisions instead of large incisions. Some of the benefits are - Less pain and blood loss, Quicker recovery, less noticeable scars. Some of the drawbacks are that it can cause bleeding and infection. Surgical robots stand out in comparison to conventional endoscopic surgeries because they provide 3d vision, wide range of motion and ability to change the scale of the robot's movement simplifying complex movements like suturing and knot tying. Surgical robots help doctors perform many complex procedures with more precision, flexibility and control.

Haptics generally means simulating sense of touch by applying forces, vibration or motion. In case of a laparoscopy surgery, Haptic feedback plays a pivotal role during the manipulation of tissues as it provides the surgeon a sense of touch as if his own hands are contacting the patient. Haptics is a combination of force feedback (form and shape of muscles, tissues and joints) as well as tactile perception (relating to the skin). The force feedback system measures the amount of force applied during the treatment of tissues in a surgery to the patient by the surgeon using surgical robot.

Surgical forceps are used to compress and grasp tissues in surgical operations. There are many surgical forceps designed with force feedback capability for minimally invasive surgery. But these forceps have not been installed in the current surgical robots because they are distorted by friction forces and reaction forces. There are physical constraints with these force sensors that hinder the sense of contact no matter what sensing algorithm is utilized.

During the surgery, haptic feedbacks allows the surgeon to glean about the mechanical properties of the tissue and further apply appropriate forces for safe mobility of the tissue. As we have established the importance of haptic perception, this report will emphasize on the force feedback estimation of a surgical robot with the external environment using concepts and application of Kalman filter. The effectiveness of this technique is shown using a one degree of freedom robot.

The grasping force control estimation of the surgical forceps is measured mainly in two ways:

Sensor based approach –

Sensor based approach involves the use of mechanical and electrical sensors. Some of the sensors that are integrated with the surgical robot are displacement sensors, strain gauge sensors, capacitive sensors, piezo electric sensors, optical fiber sensors etc. however, minimally invasive surgery is not cost effective and therefore very few commercially available sensors can be directly used in surgery.

Sensor-less approach –

In the sensor-less approach, force sensing ability is measured by finding the position, velocity and torque applied to the robot's actuators. Disturbance observer is used to estimate the forces. The conventional state observers (example - Luenberger observer) have sensitivity problems. The disturbance observer uses system dynamics and all the possible measurable states to estimate the internal and external disturbance. The internal and external disturbance are given as feedback to the system thereby making the system robust.

Another approach to estimate the force control is the use of cable driven robots. System dynamics and square root unscented unscented Kalman filter is used. The Unscented Kalman filter is used to linearly approximate a nonlinear system. Unscented Kalman filter undergoes unscented transformation where in it collects statistics on data that is going through nonlinear transformation by continuously estimating the distribution statistics which is the mean and covariance matrix of the underlying gaussian distribution. It has a set of sigma points that are chosen in such a way that they capture the true mean and covariance when propagated through the nonlinear system.

However, force estimation approach was realized to be inaccurate because of the friction in the transmission and cable properties. In order to solve this issue, four gaussian process regression filters were used. The gaussian regression process calculated the probability distribution of the function by training hundreds of data points that were able to learn models for strongly nonlinear applications. The drawback was the computation time it took when it was trained with data points. Another force sensing estimation method was the use of screw theory, Lagrange dynamics and rotor dynamics but the parameters used to estimate the forces were sensitive to the proposed model.

All the force estimation method mentioned above involved the use of torque observers. The actuation unit used in the surgical robot is the DC motor which has the following features –

- Low moment of inertia which minimizes torque requirements for the application
- No cogging torque which is produced due to interaction between rotor and stator slots of a permanent magnet. This is an undesirable component because it results in jerkiness and speed ripple
- Low friction
- Very compact commutation which is the process of converting alternating current generated in the armature winding of the DC motor into direct current after going through commutator and the stationary brushes.

These features of the DC motor result in –

- High acceleration
- High efficiency
- Low joule losses
- Higher continuous output torque

The surgical robots do not require high joint speeds and therefore the actuation unit that is used is a small, low power motor with higher gear reduction. Gear reduction is done by reducing the rotational speed of the rotary machine by dividing it by the gear ratio greater than 1:1. A gear ratio greater than 1:1 is achieved when the smaller gear (reduced size) with fewer number of teeth meshes drives a larger gear with greater number of teeth.

Surgical forceps with highly geared drive mechanism are either non back drivable or difficult to back drive. Back drivability is the ability for interactive transmission of forces between the input axis and output axis. Back drivable component is used in reverse direction to obtain its input from the output. Not every system is back drivable. To achieve high back drivability, we must reduce friction in power transmission considerably.

Motor current cannot measure the external applied force in stationary non back drivable system because the external contact force is not transmitted to the driving motor. The driving torque of the DC motor includes the external contact force, noise torque and load torque caused by gear reduction and components of the robot. Therefore, DC current cannot accurately estimate the external contact force.

Section 2:

Kalman filter is used in this project to predict the driving current of the DC motor, as the current signal of the DC motor is noisy, and a specially designed current sensor is required for current measurement. The external contact force on the end joints of the effector is estimated by the current of the DC motor, as the torque is a function of current and the torque constant, as explained later. The estimated contact force of the end joint is further compared with the actual measurement from the force sensor.

Estimation Methods:

For force estimation, we are going to start by asserting that torque in DC motor is proportional to current in the motor servo amplifier. Figure attached below presents the model of DC motor regulated under current source with Kalman filter.

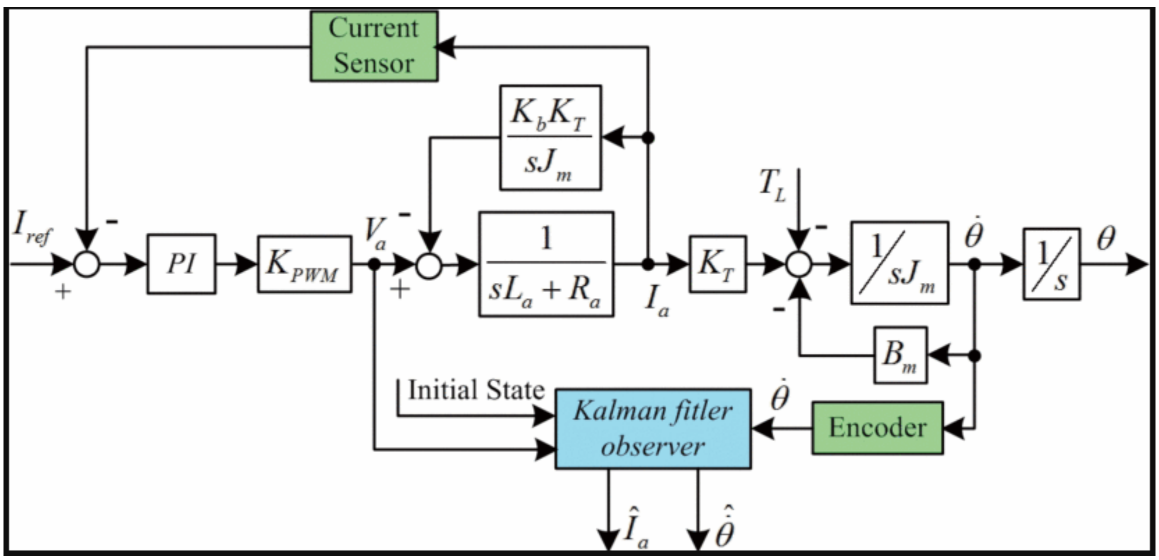

Fig 2.1 Model of the linear DC motor regulated under the ideal current source with a Kalman filter.

Since this is a sensor less approach for force estimation, we must try to find the current value without using any other sensor. However, it is useful if we have a pre-installed transducer, such as the position sensor, to be utilized.

There are 3 controllers being used in this image. A PI controller is used to compare the current values and the resultant is fed into the motor driver that controls the motor. A controller is used to compare the values of voltage being fed to the motor with the value obtained from the Kalman filter. A high-resolution optical encoder is used to measure the position of the end joint. This position signal acts as the input signal to Kalman filter, along with the driving voltage of the robot handle, for estimating the driving current and velocity.

A. Armature Controlled DC Motor

An armature-controlled DC motor is employed in this study to perform state prediction. We are going to derive the transfer function for the DC motor in the next few steps.

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in the figure below.

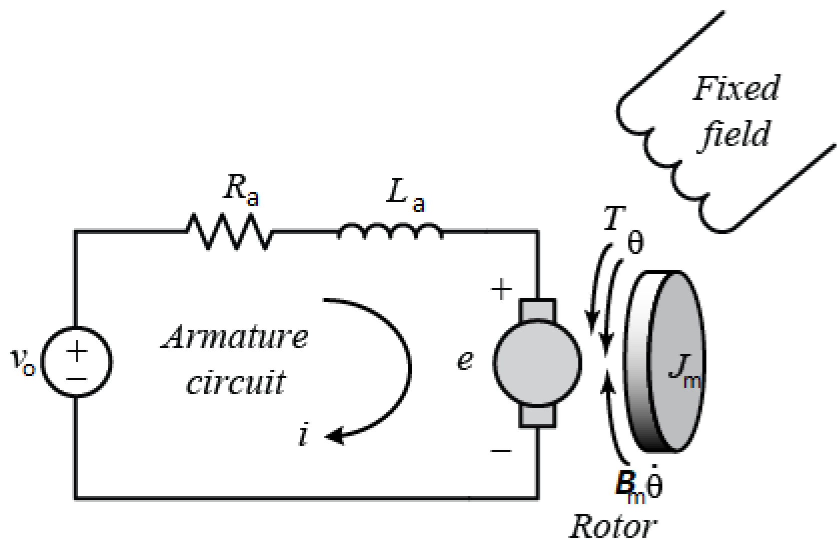

Fig 2.2 Electric equivalent circuit of the armature and the free-body diagram of the rotor

The input of the system is the voltage source V_o applied to the motor's armature, while the output is the rotational speed of the shaft $\dot{\theta}$. The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.

The torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. The magnetic field is constant and so, the motor torque is proportional to the armature current i by a constant factor K_t. This armature-controlled motor is represented by the equation.

$$T = K_t i$$

The back emf, e, is proportional to the angular velocity of the shaft by a constant factor K_b.

$$e = K_b\dot{\theta}$$

Based on Newton's second law, we come up with the torque equation as follows:

$$J_m\ddot{\theta} + B_m\dot{\theta} = K_t i$$

$$L_a\frac{di}{dt} + R_a i = V_o - K_b\dot{\theta}$$

Applying the Laplace transform for these 2 equations, we obtain the following equations:

$$s(J_m s + B_m)\,\theta(s) = K_t I(s)$$

$$(L_a s + R_a)I(s) = V_o(s) - K_b\theta(s)$$

Eliminating $I(s)$ from the last 2 equations obtained and obtaining the transfer function:

$$\frac{\theta(s)}{V_o(s)} = \frac{K_t}{s[(L_a s + R_a)(J_m s + B_m) + K_t K_b]}$$

To clarify everything in the above equation:

where θ is the position of the motor shaft, V_o is the applied voltage, L_a is the armature winding inductance, R_a is the armature winding resistance, J_m is the moment of inertia of rotor and load, B_m is the damping coefficient, K_t is the motor torque constant, and K_b is the back emf constant.

Now, as per the report, driving current and velocity are chosen as the state variables. So, the state equations are given as follows:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_b}{L_a} \\ \frac{K_t}{J_m} & \frac{-B_m}{J_m} \end{bmatrix}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} u$$

$$y = [0 \;\; 1] \begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix}$$

Where, X = $[i_a \quad \dot{\theta}]^T$ and $\dot{\theta} = \frac{d\theta}{dt}$. In measurement, the rotating shaft of the driving motor is the output **Y** and input **U** is V_o.

For a continuous system, the general form of the state equation is given as follows:

$$\dot{V}(t) = A_c V(t) + B_c U(t)$$

$$Y(t) = C_c V(t) + D_c U(t)$$

Where *V(t)* is the variation of the estimated rotating speed. A_c, B_c, C_c, D_c are the coefficient matrices of the state equation for a continuous system. Now, we can compute the state transition matrix as follows: $\emptyset_c = L^{-1}[(sI - A_c)^{-1}]$.

The discrete state equations sampled from the above equations by a sample-and-hold with time interval of T seconds is given as follows.

$$X_{k+1} = A_d X_k + B_d U_k$$

$$Y_k = C_d X_k + D_d U_k$$

Where A_d, B, C_d, D_d are the coefficient matrices of the state equation for the discrete systems, which are calculated as follows

$$A_d = \emptyset_c(t); \; B_d = [\int_0^T \emptyset_c(\tau)d\tau] \, B_c; \; C_d = C_c; \, D_d = D_c$$

B. Kalman Filter for state estimation

Kalman filter is a type of state observer that is designed for stochastic systems (there is some degree of randomness associated with the system, unlike a deterministic system). It is a commonly used optimal linear observer, which estimates the system state, given a system model, input, and measurement. Kalman filter minimizes the inconsistencies with all information available to obtain the best possible estimation. State estimates and measurements are expressed as Gaussian distributions, express a desired level of confidence on the information.

Let us assume that we want to know the value of a variable within a process of the form:

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

where; x_k is the state vector of the process at time **k**; A_k is the state transition matrix of the process from the state at **k** to the state at **k+1**, and is assumed stationary over time; w_k is the associated white noise process with known covariance. Observations on this variable can be modelled in the form:

$$y_k = h(x_k) + v_k$$

where; y_k is the actual measurement of x at time **k.** h is the noiseless connection between the state vector and the measurement vector and is assumed stationary over time. v_k is the associated measurement error. This is again assumed to be a white noise process with known covariance and has zero cross-correlation with the process noise. Both w_k(process noise) and v_k(observation noise) have Gaussian distributions.

It is assumed that the covariance matrix of the process noise **Q**, covariance matrices of the measurement noise **R**, and cross-covariance matrix are:

$$Q=E\{\omega\omega^T\}>0$$

$$R=E\{vv^T\} \geq 0$$

$$E\{\omega v^T\}=0$$

The bandwidth of the Kalman filter and its susceptibility to measurement noise all depend on its process noise covariance matrix **Q** and the measurement noise covariance matrix **R**. We will use these values to tune the accuracy and response of the observer.

Let us define an error covariance matrix $P_k$at time k as follows.

$$P_k = E[e_k \, e_k{}^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

Assuming the prior estimate of $\hat{x}_k$ is called $\widehat{x`}_k$ and was gained by knowledge of the system. It possible to write an update equation for the new estimate, combing the old estimate with measurement data thus;

$$\hat{x}_k = \widehat{x`}_k + K_k(y - h\widehat{x`}_k)$$

Where $\widehat{x`}_k$is known as the 'priori estimate' and $\hat{x}_k$ is known as the 'posteriori estimate' and K_k is called the Kalman gain. A priori estimate predicts the current state by using the state estimate from the previous time step and current input. The second part of the posteriori estimate uses measurement and incorporates it into the prediction to update a priori estimate.

The measurement and state equations are linearized about the most recent state estimate in the Kalman filter. The two main steps constitute the filtering are: prediction and update. Kalman filters recursive algorithm can be represented as follows:

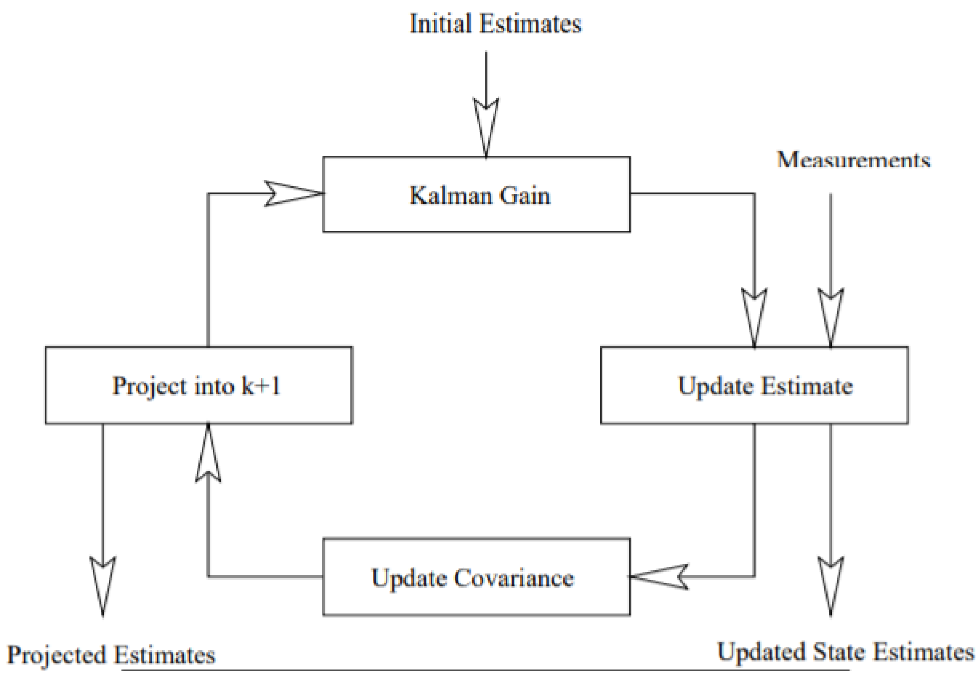

Fig 2.3 Kalman filter recursive algorithm

Representing Kalman gain in terms of equations, we get the following:

Description	**Equation**
Kalman Gain	$K_k = P_k^{`}\, H^T (HP_k^{`}H^T + R^{-1})$
Update Estimate	$\hat{x}_k = \widehat{x^{`}}_k + K_k(y - H\widehat{x^{`}}_k)$
Update Covariance	$P_k = (I - K_k H)\, P_k^{`}$
Project onto k+1	$\hat{x}_{k+1} = A_k \hat{x}_k$ $P_{k+1} = A_k P_k {A_k}^T + Q$

Table 2.1 Representing Kalman gain in terms of equations

Now, we try to adapt the general Kalman equations to the way the authors have chosen in their paper.

For prediction, the state transition matrix A_{k-1} is computed as the Jacobian of the state function evaluated at the previous state estimate (k-1). In the update stage, the observation matrix H is computed as the Jacobian of the function $h(\cdot)$ with respect to the state vector x_k, where $(\check{\cdot})$ and $(\hat{\cdot})$ are the prediction and estimates. The subscript $k|k-1$ is the computations at time k, given others at time k−1 and similarly for $k-1|k-1$ and $k|k-1$.

Filtering algorithm based on Gain prediction is given as follows:

1. **State Prediction**
   We are going to compute the state prediction and covariance matrix P calculation:
   Computing the state prediction matrix with respect to the previous time step:
   $$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_k$$
   The covariance matrix is calculated based on the previous time step as follows:
   $$P_{k|k-1} = A_{k-1} P_{k-1|k-1} {A_{k-1}}^T + W_{k-1}$$
   Where, $W_{k-1}$is the gaussian white noise matrix associated with process noise.
2. **Gain adjustment**
   Let us define a new operator S, which is called the gain scheduling operator. The covariance matrix of the prediction error is related to the covariance matrix by the following relation:
   $$\sum_{k|k-1} = S(P_{k|k-1})$$
3. **Measurement Innovation**
   Now, the authors update the prediction measurement matrix as follows:
   $$\hat{Y}_k = Y_k - H_{k-1} \hat{X}_{k|k-1}$$
   The estimate covariance matrix is now formulated as follows:
   $$\tilde{P}_k = H_{k-1} \sum_{k|k-1} {H_{k-1}}^T + V_k$$
   Where, $V_{k-1}$is the gaussian white noise matrix associated with measurement noise.
4. **Estimation update**
   The filtering gain and estimation of state using the covariance matrix is given by the following equations:
   $$K_k = \sum_{k|k-1} H_k^T \tilde{P}^{-1}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \hat{Y}_k$$

$$P_{k|k} = (\sum_{k|k-1}{}^{-1} + H_k^T V_k^{-1} H_k)^{-1}$$

Section 3:

System Dynamics

In the previous section we discussed the usage of a Kalman filter to estimate the end effector velocity. Now we use the end effector measurements to calculate the values of end effector forces and torques.

The robot force and torque are estimated by modelling the robot dynamics. The robot dynamics and relationship between the force and torque on the end effector and joint torque on the robotic is defined by the following equation:

$$\tau_{ext} = J^T F$$

Where J is the robot Jacobian, F is the force vector at the end effector. Now, we calculate the torque vector of the DC motor exerted in each robot joint as follows:

$$\tau_m = J_m(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\dot{\theta} + C(\theta) + \tau_f(\dot{\theta}) + \tau_{ext}$$

Where $J_m(\theta)$ is the robot inertia matrix , $H(\theta, \dot{\theta})$ is the Coriolis forces vector, $C(\theta)$ is the gravity force vector, $\tau_f(\dot{\theta})$ is the friction vector torque and τ_{ext} is the external torques on each joint produced by external force on the end effector. Now, we use the last equation to represent τ_{ext} in the first equation to find out force vector matrix.

$$F = (J^T)^{-1} \left(\tau_m - J_m(\theta)\ddot{\theta} - H(\theta, \dot{\theta})\dot{\theta} - C(\theta) - \tau_f(\dot{\theta})\right)$$

For a back drivable robot, the external force on the robot tool tip can be estimated by the driving current in the motor driver. Given the motor torque constant K_t and the current amplitude I_a, we can calculate the value of torque vector of the DC motor.

$$\tau_m = K_t I_a$$

Section 4:

Experimental Setup

Figure attached below describes the experimental setup the authors used for their demonstration.

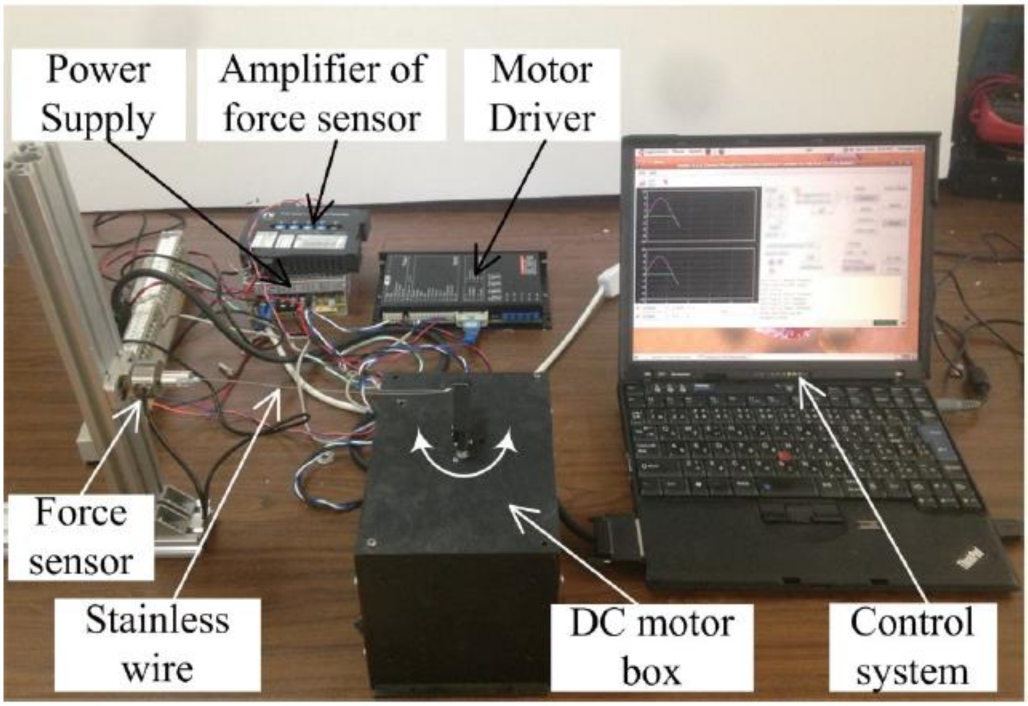

Fig 4.1 Experimental set up for sensor-less force estimation

This is a setup for sensor-less force estimation. It consists of the following hardware materials:

1. Linux OS based real time control system. More details regarding this system is described in the following sections.
2. 24V DC power supply.
3. Maxon RE 25 DC motor, specs discussed later.
4. Force sensor amplifier.
5. Force sensor with stainless wire and load cell (ZNLBM-1KG) and transmitter (DMD4059).
6. Optical encoder (MEH-12-1000P)
7. Amplifier sensor PWM amplifier (MAXON ADS 50-2)
8. Motor driver setup.

A close-up view of the experimental setup's transducer part is shown below:

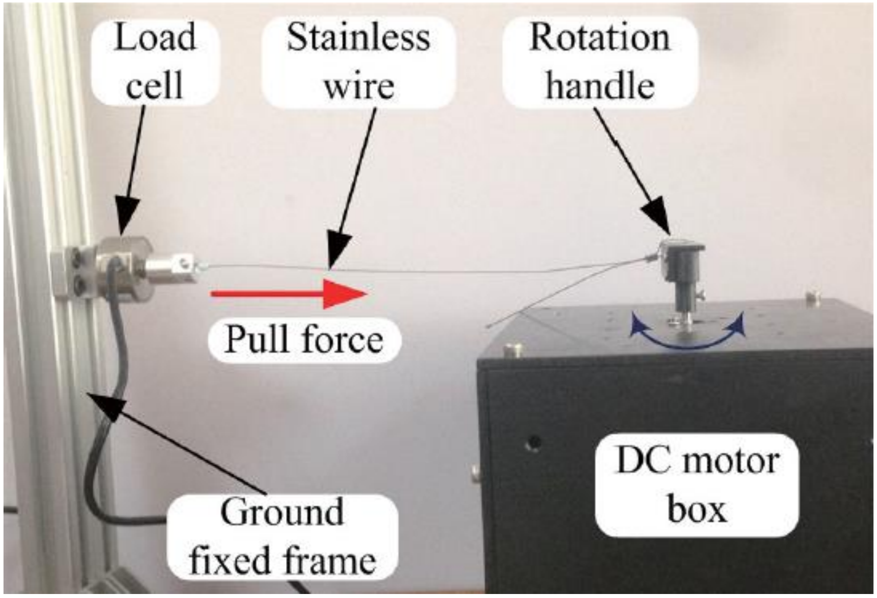

Fig 4.2 External contact force measurement set up

The data acquisition system is described later in this section.

A. The driving DC Motor

The DC motor used in this experiment was made by Maxon, Switzerland which was driven by a PWM Amplifier (Maxon ADS 50-5). The model number of the motor is RE 25.

Parameters for the DC motor used in this study are as follows:

1. DC Voltage for the motor, V_o: 24V
2. Damping coefficient, B_m: 2 x 10^{-6} Nms.
3. Armature winding inductance, L_a: 8.3 x 10^{-4} Henries
4. Armature winding resistance, R_a: 7.31Ω
5. Moment of inertia for the motor and load, J_m: 1.05x10^{-6} kgm^2
6. Motor torque constant, K_t= 0.044 Nm/A
7. Back emf constant, K_b= 22.7 rad/Vs

Framing the state space model using these parameters on MATLAB, we get the following matrices.

For a continuous system:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -8.807x10^3 & -2.7349x10^4 \\ 4.1905x10^4 & -2 \end{bmatrix}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 1204.9 \\ 0 \end{bmatrix} u$$

$$y = [0 \;\; 1] \begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix}$$

We notice that there is a small conflict with the research paper at A_{12}. It says -52.8, which is not possible for the given values of K_b and L_a.

After research we found out that in SI units, the value of K_t and K_b are the same, that is the motor torque and back emf constants are equal. Therefore, we proceed with the value of $K = K_t = K_b$ = 0.044Nm/A.

Therefore, the new state space matrix would be as follows:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -8.807x10^3 & -53 \\ 4.1905x10^4 & -2 \end{bmatrix}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 1204.9 \\ 0 \end{bmatrix} u$$

$$y = [0 \;\; 1] \begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix}$$

Which is the same as the matrix obtained in the paper.

We create a state space model of the DC Motor in MATLAB by initializing the A, B, C, D matrices with time interval Ts = 0.001s as it is a continuous state.
Here, the Input name is given by V, the output name is given by Y, and the state name is given by i_a and $\dot{\theta}$.
We proceed towards calculating the discrete state equations sampled from the above equations with time interval T=0.001s.

B. Data Acquisition system

A real time control system based on Linux OS was installed and used. Reason was to maintain accurate sampling. The controller of the one DOF system is implemented at the PC. The target PC contained an AD/DA multi-function card (CSI-360116, Interface Co., Ltd) for Card Bus systems with differential A/D lines for reading the load cell, and D/A for outputting command signals to the motor amplifier. This card was also be used to read output pulse of the encoder signals from the optical encoder rotation sensor. The core controlling program ran for the duration of a block of trials, reading sensors and commanding motor torque at a rate of 1 kHz through experimental conditions. An optical encoder (MEH-12-1000P, Micro tech Laboratory Inc.) is used at the end axis of the DC motor to measure the rotational position of the handle. Velocity is not directly measured; it is obtained by differentiating the rotary position with respect to time.

The external contact force is measured by a load cell (ZNLBM-1KG, Zhongnuo Co., Ltd) with a transmitter (DMD4059, Omega Co. Ltd).

Section 5:

Results and Discussion as observed from the paper:

From their experimental setup, the following results were obtained. For the given reference input signal, they have plotted graphs and analytically compared the kinesthetic performance of the proposed current based external force estimation method. Experimental evaluation was also done on a later stage.

Using our basic understanding, we interpret the results of the graphs as follows:

A. Test position tracking during free motion/Free motion with Kalman filter estimation:

In this case, there is no contact between the robot end effector and the environment, i.e. there is no constraint. Driving parameters for the handle was as follows: A sinusoidal reference signal with an amplitude of 6 deg @ 0.4Hz frequency. The experimental observations are shown below for the performance of the robot end effector estimated using a Kalman filter and comparing it with a reference signal.

1. Position tracking performance:

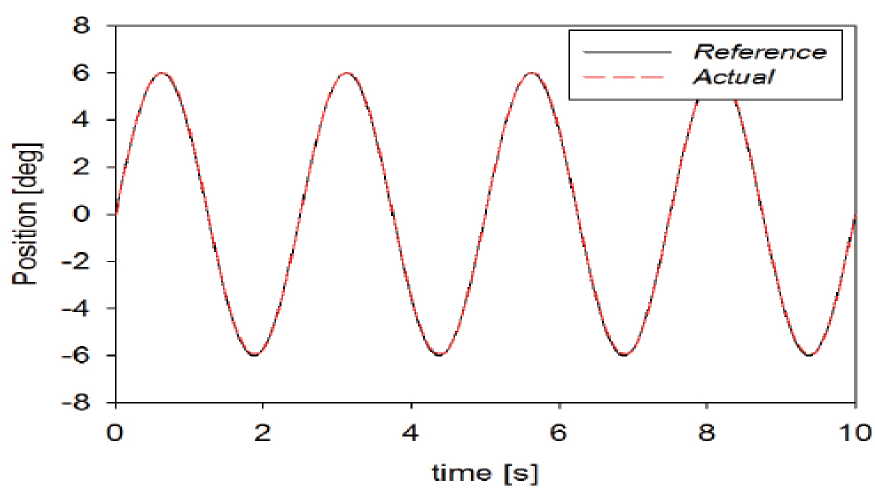

Fig 5.1 Position (deg) vs Time (s)

We see that the reference and the actual signal for position are overlapping here. This is confirmation that at no contact condition, the robot tracks reference position signal very well.

2. **Estimated velocity by the Kalman filter:**

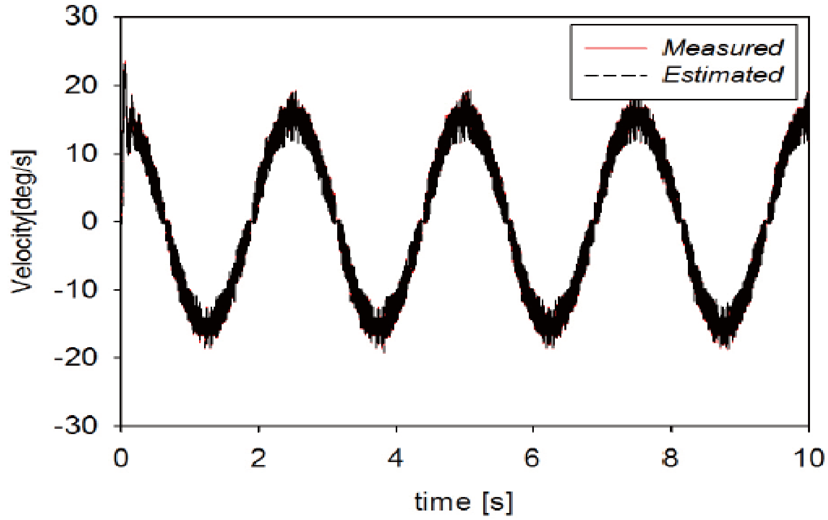

Fig 5.2 Velocity (deg/s) vs Time (s)

Here, we observe that again, estimated velocity is overlapped by the actual velocity and both are functionally equal with minimum errors.

3. **Actual motor current graph:**

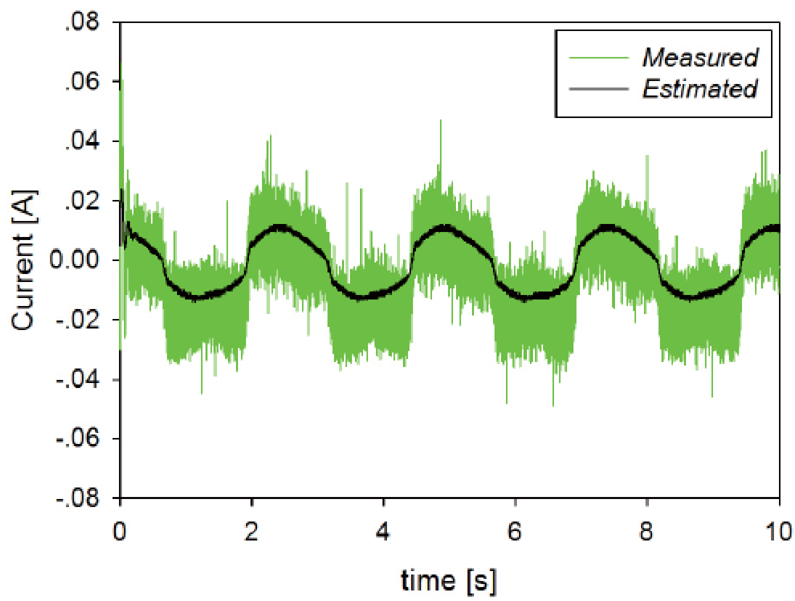

Fig 5.3 Current (A) vs Time (s)

Over here, we observe that the Kalman filter estimated current is free from noise and has a slight advantage over the measured current.

4. **Force estimation in contact less condition:**

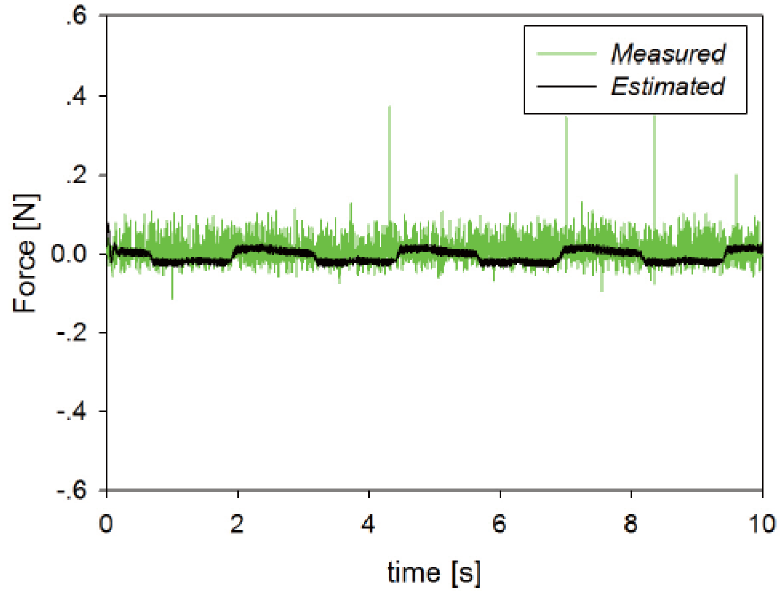

Fig 5.4 Force (N) vs Time (s)

As expected for non-constrained motion, the force at the end effector position is very close to zero in both the conditions. However, Kalman filter estimated force is more accurate and is devoid of white noises, unlike the measured values.

B. **Test position tracking during constrained motion/External force estimation by Kalman filter:**

Now, the robot handle is constrained by the stainless-steel wire of diameter = 0.4mm (AB018-50FT, 7x7 strands, Berg Inc.) The rest of the setup is the same as shown in the experimental setup diagram. Calculated current from Kalman filter is used to estimate the force at the robot end effector. Here, we compare the accuracy of external force estimation by the output of the load cell.

As in the previous case, driving parameters for the handle was as follows: A sinusoidal reference signal with an amplitude of 6 deg @ 0.4Hz frequency. The experimental observations are shown below for the performance of the robot end effector estimated using a Kalman filter and comparing it with a reference signal.

1. **Position tracking performance:**

This result is as expected. The reference signal is a complete sinusoidal wave with a fixed amplitude and frequency. However, the actual position of the motor is constrained in one half but manages to track the reference signal in the other half.

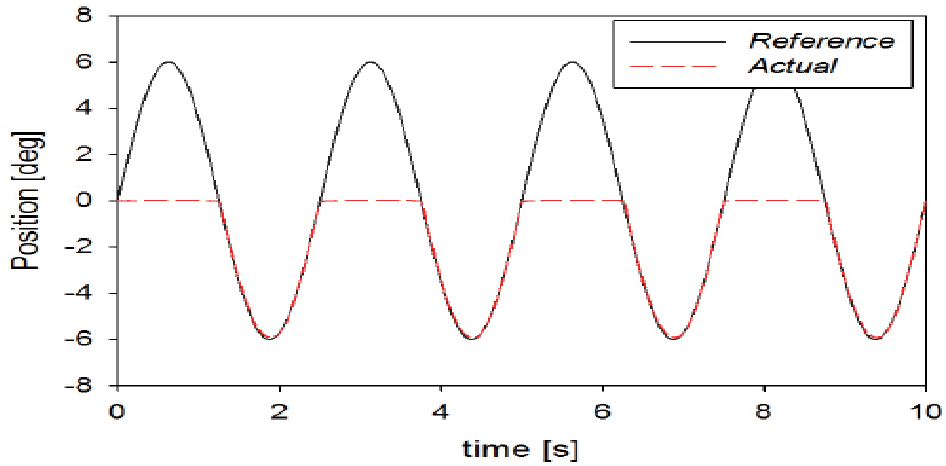

Fig 5.5 Position (deg) vs Time (s)

From this observation, we can understand how the robot behaves in a constrained environment. In the first half, the robot is unable to move from its mean position as it is completely constrained by the stainless wire and in the second half of the signal, it can track the reference signal appropriately. Also, notice that the robot handle pulls the string that is further connected to a load cell, that is used to compute the force values.

2. **Estimated velocity by the Kalman filter:**

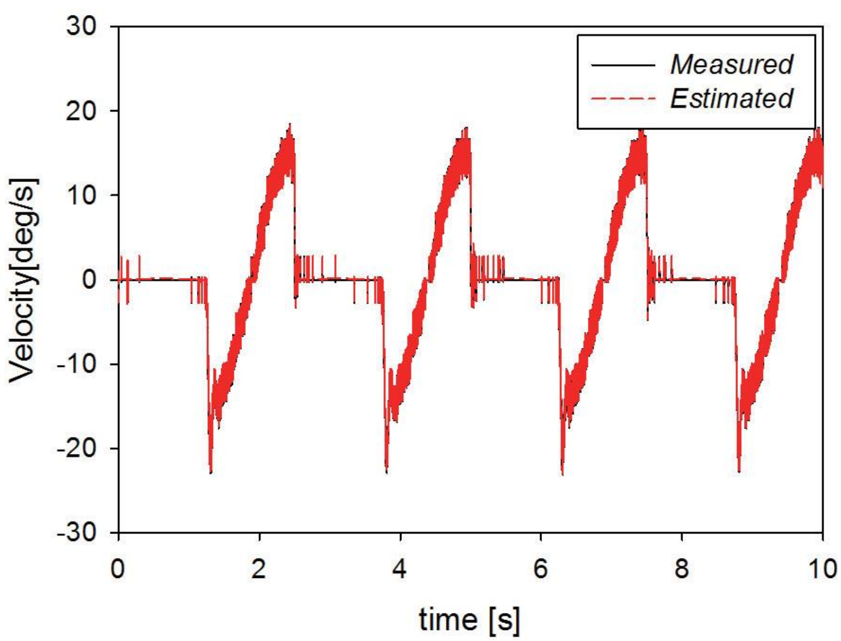

Fig 5.6 Velocity (deg/s) vs Time (s)

Here, we observe that Kalman filter estimation of the of the measured velocity accurately traces the measured values as observed from the setup. An optical encoder is used to calculate the velocity of the end effector.

3. **Actual motor current graph:**

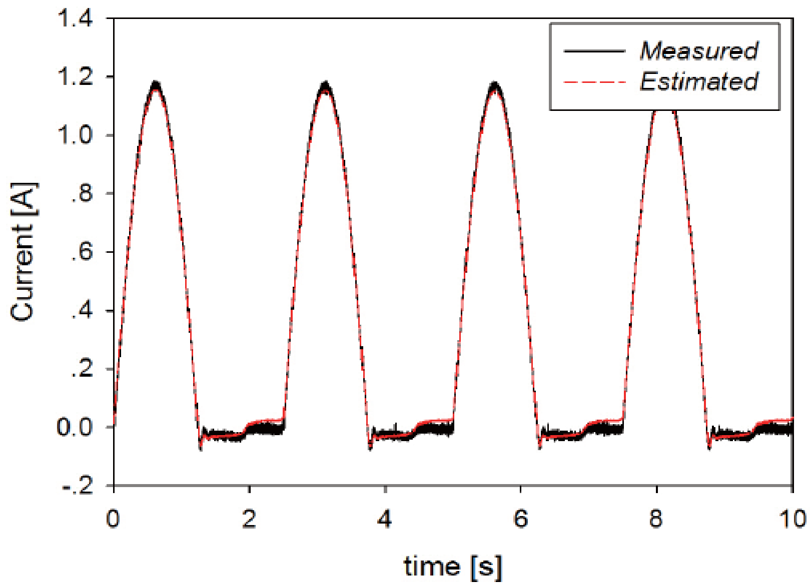

Fig 5.7 Current (A) vs Time (s)

Again, we can observe that the Kalman filter estimated current closely follows the measured current taken using a current sensor in the experimental setup.

4. **Force estimation in contact less condition:**

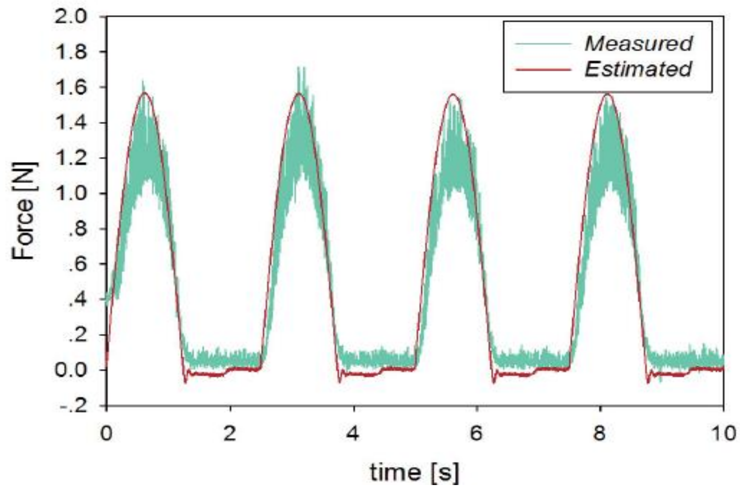

Fig 5.8 Force (N) vs Time (s)

Here, we see that the Kalman filter estimated force is rather accurately tracing the value of the measured force. We use the current output of the Kalman filter (as shown in the previous graph) to calculate the force values at the end effector positions by the formulas discussed in above sections.

Therefore, we can conclude that Kalman filters can be used in both contact and no contact conditions to accurately estimate external force of the end effector.

Results and Discussions based on what we have simulated:

The results shown in the research paper has a setup that is hard to realize in a laboratory on campus. So, we have used Simulink and Simscape to model a DC motor and carry out the simulations on them.
We create a state space model of the DC Motor in MATLAB by initializing the A, B, C and D matrices with time interval Ts = 0 as it is a continuous state. The computed A, B, C and D matrices have the following values (as computed earlier in MATLAB).

$$A= \begin{bmatrix} -8.807x10^3 & -53 \\ 4.1905x10^4 & -2 \end{bmatrix}$$

$$B= \begin{bmatrix} 1204.9 \\ 0 \end{bmatrix}$$

$$C=[0 \;\; 1]$$

$$D=0$$

And they satisfy the equation:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -8.807x10^3 & -53 \\ 4.1905x10^4 & -2 \end{bmatrix}\begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 1204.9 \\ 0 \end{bmatrix} u$$

$$y = [0 \;\; 1] \begin{bmatrix} i_a \\ \dot{\theta} \end{bmatrix}$$

Here, the input u, is voltage V_o which is approx. 24V for the DC motor. The output y, is a direct measure of the angular velocity, which can be further used to calculate force of end effector.

Major Learning Outcomes:

1. **Stability Analysis**

We find the poles of the system to verify whether the system is stabilizable. If the real part of the poles is negative, then the system is stabilizable. In a complex plane with Real part of the pole in X axis and Imaginary part in the Y axis, if the poles of the system lie in the left half of the complex plane, then the system is stable.
The poles are given by -

-8.5473 * 10^3

-0.2619 * 10^3

Clearly, both poles have negative real parts, hence **the system is stable**.
The eigen vectors are given by –

$$\begin{bmatrix} -0.1998 & 0.0062 \\ 0.9798 & -1.0000 \end{bmatrix}$$

2. **Controllability Analysis**

To make a system controllable for a given value of input, we must ensure controllability for the state equations. Controllability for a state space equation can be checked by the following condition:
If the matrix has linearly independent columns where the Rank Ç = n (dimension of the state) then, the system is controllable, where Ç is given by:

$$Ç = [B\ AB\ A^2B\ \dots\ A^{n-1}B]$$

Alternatively, we can use the ctrb () function, coupled with rank() in MATLAB to compute the same.
We find the controllability matrix of the continuous state system to be the following:

$$Ç = \begin{bmatrix} 10\text{^}3 & -1.0611x10^7 \\ 0 & 5.0488x10^7 \end{bmatrix}$$

On checking for the rank of this matrix on MATLAB, we observe that rank of Ç is 2, which is the same as the dimension of our state. So, this system is **controllable**.

3. **Observability Analysis**

To understand what is happening inside the hood of the system, the system must be observable. For given A and C matrices, we can compute the observability matrix on MATLAB as follows:

The observability matrix is given by:

$$Obs = \begin{bmatrix} 0 & 1 \\ 4.1905x10^4 & -2 \end{bmatrix}$$

Since the rows of the matrix are linearly independent, the rank is 2 which is a full rank implying that the system under consideration is **observable**.

In order to amplify the input voltage, voltage amplifier is used with the following parameters –

- Steady state voltage gain – 24 (V)
- Bandwidth 2500 (Hz)

Voltage amplifier is represented in MATLAB using first order transfer function.

Now, we create the overall plant model by combing the state space model and the voltage amplifier using MATLAB.

As we have determined the stability of the system, we will analyze the step response for a unit step voltage applied using the step function in MATLAB.

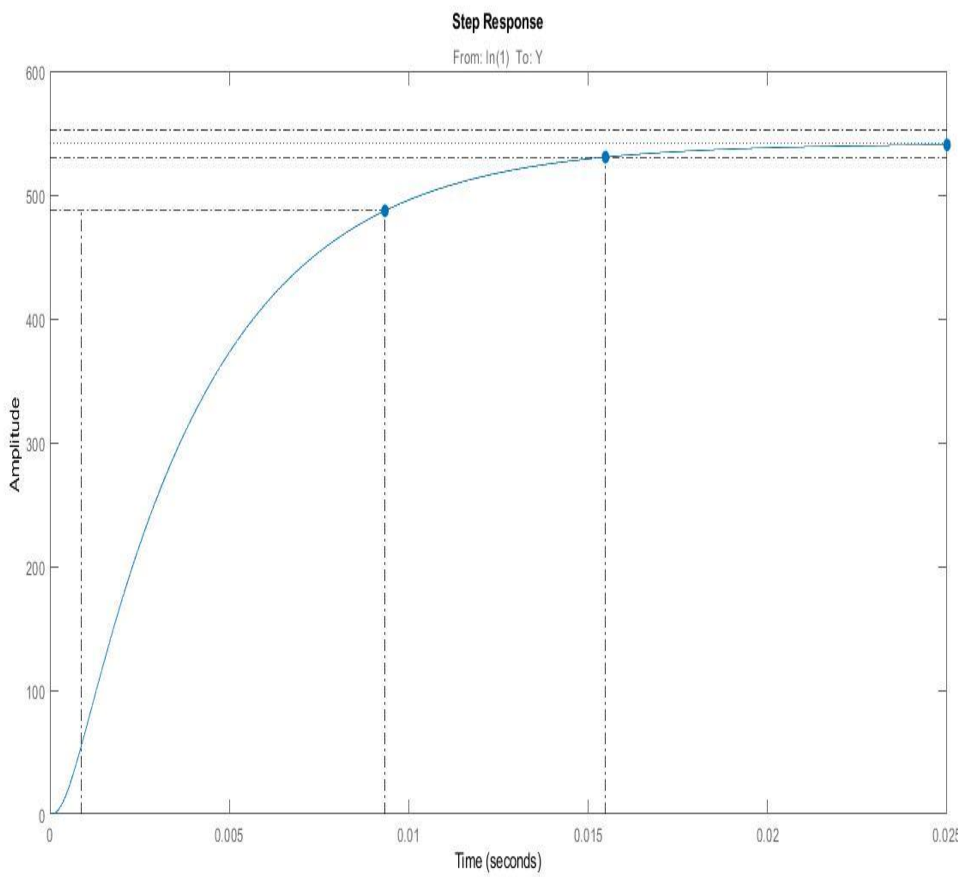

Fig 5.9 Amplitude vs Time (s)

From the transient system characteristics, we can see the **rise time**, **the settling time** and the **steady state**.

Apart from this, we have created a Simulink Model of the DC Motor and have assigned them the given parameters as explained in the paper. The DC Block for the same is as shown below:

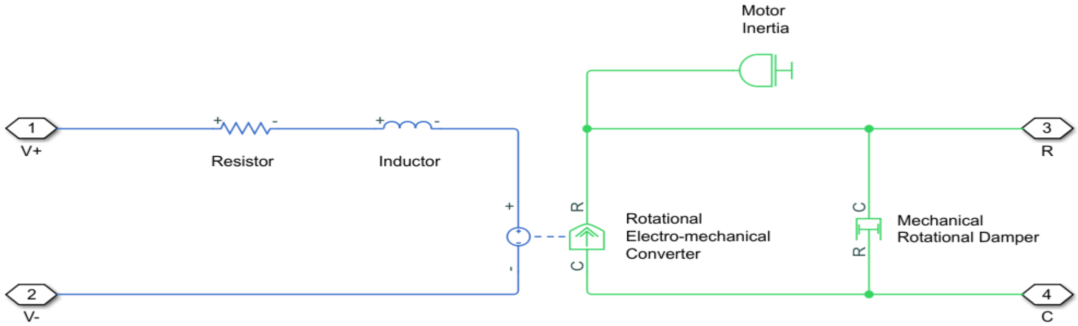

Fig 5.10 Model of DC motor in SIMSCAPE

The values of inertia, resistance and inductance are taken from the DC motor specification sheet. The motor model in SIMSCAPE is modelled as below:

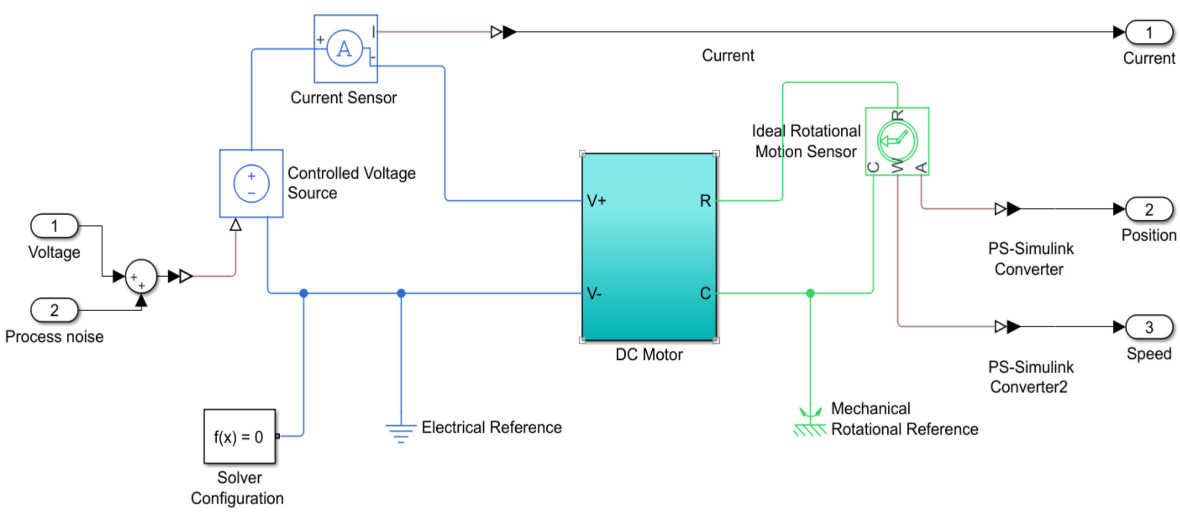

Fig 5.11 Model of DC Motor in Simscape – connected to other electrical sensors

Now, we provide power source and introduce some process noise into our system and observe the outputs, in terms of speed, current and position. Each of these outputs are further mixed with some measurement noise and the output is now fed into a Kalman filter to remove the noise at process and measurement level, by directly comparing with the input voltage source as shown.

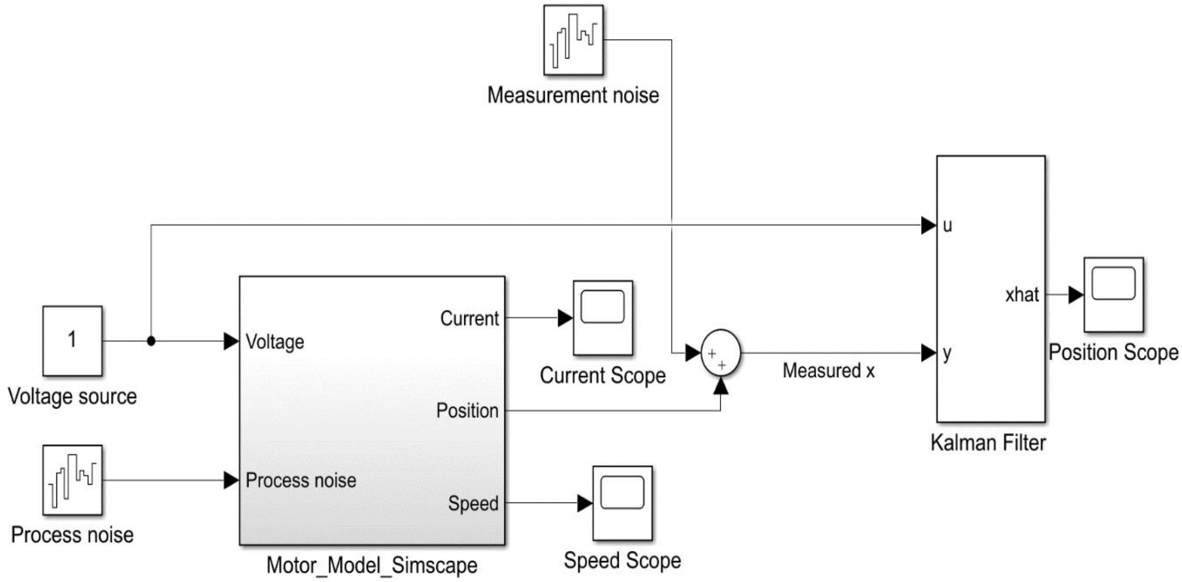

Fig 5.12 DC motor simulation connected to a Kalman filter for position estimation case

We carried out the results for a constant voltage source as shown in the images shown above. The Kalman filter is made for a continuous time system with the same state space properties as mentioned in the research paper. The values of process noise covariance and Measurement noise covariance have been appropriately assumed, as the values were not mentioned anywhere in the research article. We obtained the following result for a constant input source of 24V for the DC motor.

When we observe the current scope in the image shown above:

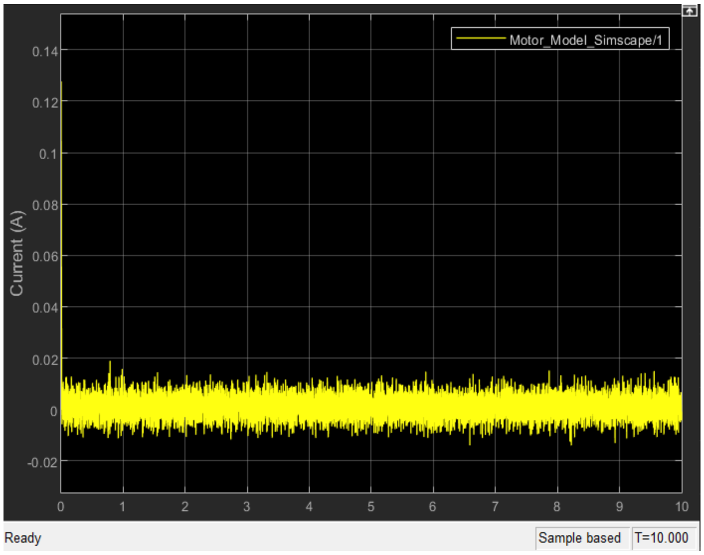

Fig 5.13 Current (A) vs Time (s)

Which is relatively the expected value for current with process noise involved, for a constant DC voltage supply.

Now we observe the speed plot. Notice that the speed port is not connected to the Kalman filter in this case, so whatever we are getting is the measured speed output for the same.

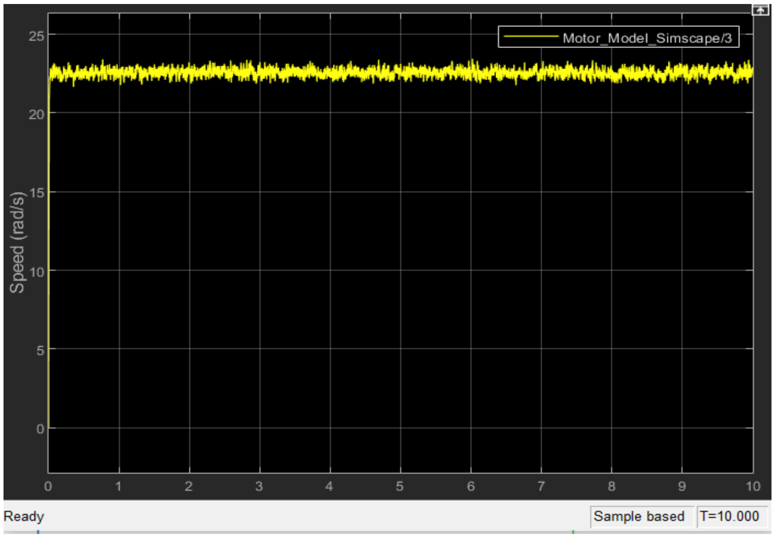

Fig 5.14 Speed (rad/s) vs Time (s)

For a given steady constant DC voltage, we get a constant speed spread about because of the noise matrix.

Now, we check the position graphs. The output from the DC motor is then sent to the Kalman filter along with the measurement noise. This is as shown in the Simscape model shown above. We obtain this graph.

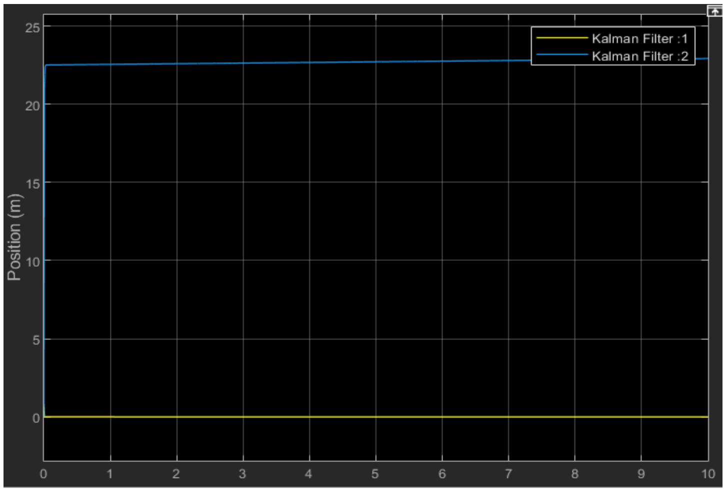

Fig 5.15 Position (deg) vs Time (s)

These are the results we obtained for a constant DC voltage source. We are hoping to improve on the result accuracy in the next few days, as discussed below.

Section 6:

Conclusion

Ways to deal with control of external contact forces of surgical manipulators have gotten expanded consideration in the improvement of Minimally Invasive Surgical robot system. To maintain a strategic distance from the inconveniences of force sensor installation, a discretionary technique that guarantees the robustness and contact stability of a feedback control system is displayed in this examination. The external contact force was assessed by the driving current of the motor. A Kalman filter disposes of force noises accomplishing stable and dependable position responses. The external force and device movements can be indistinguishably pursuing the tracks of the actual contact force and movements. Besides, the proposed external force estimation approach can intertwine the data of the optical encoder and current sensor. The proposed external force estimation approach might be the key innovation in future MIS robot system. The proposed strategy can be applied to some other domains, for example, haptic device with reliable force feedback.

Future Scope and what we plan to do:

Based on our simulation of the robot end effector system, we were able to identify that the system is Stable, controllable and observable. We further created models of the DC motor on Simulink and Simscape based on the values that was obtained from the research paper and available motor specifications. However, we had a few shortcomings while simulating the entire model to compare it in a discrete time domain and against the reference signal of amplitude 6 degrees and 0.4Hz frequency. With the models that we have created and with our instructors' input and feedback, we are sure to make significant progress in the same project and get comparable results. The importance of Kalman filters in accurate force estimation was conveyed through the results of the experimental setup here and we were able to successfully understand the content of the research paper. Simulation is where we are hoping to improve on and close on our report.

Section 7:

Bibliography

1. K.Ohishi, M.Miyazaki, and M.Fujita, “Hybrid control of force and position without force sensor,” in Proc. IEEE International Conference on Industrial Electronics, San Diego, CA, USA,1992, pp. 670-675
2. H.Li, K.Kawashima, K. Tadano,S. Ganguly,S. Nakano, "Achieving haptic perception in forceps’ manipulator using pneumatic artificial muscle," IEEE/ASME Transactions on Mechatronics, vol.18, no.1, pp.74 85, 2013.
3. B.Zhao, and C.A.Nelson, "Sensor-less force estimation for a three degrees-of-freedom motorized surgical grasper," Journal of Medical Devices, vol.9, no.3, pp. 030929-030929-3, 2015.
4. Y.Li, M.Miyasaka, M.Haghighipanah, L.Cheng, and B.Hannaford, “Dynamic modeling of cable driven elongated surgical instruments For sensor-less grip force estimation,” in Proc. IEEE Int. Conf. Robot. Autom., 2016, pp.4128-4134.
5. M.Haghighipanah, M.Miyasaka, and B.Hannaford, "Utilizing elasticity of cable driven surgical robot to estimate cable tension and external force," IEEE Robotics and Automation Letters, Preprint Version, pp.1-8, 2017.
6. Y.Li,and B.Hannaford, "Gaussian process regression for sensor-less grip force estimation of cable-driven elongated surgical instruments," IEEE Robotics and Automation Letters, vol.2, no.3, pp.1312-1320, 2017.
7. H.Sang, J.Yun, R.Monfaredi, E.Wilson,H. Fooladi, K.Cleary, "External force estimation and implementation in robotically assisted minimally invasive surgery," The International Journal of Medical Robotics and Computer Assisted Surgery, Early View, pp.1-15, 2017.
8. U.Kim, D.-H.Lee, H.Moon, J.C.Koo, and H.R.Choi, “Design and realization of grasper-integrated force sensor for minimally invasive robotic surgery,” in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp.4321-4326.

```
%PROJECT CODE 1
clc
clear
Ra = 7.31; % Armature Winding Resistance in ohms
La = 8.3*10^-4; % Armature Winding Inductance in H
Kb = 22.7; % Back emf constant in rad/Vs
Kt = 0.044; % Motor Torque Constant in Nm/A
Jm = 1.05*10^-6; % Moment of Inertia of the Rotor in kg*m^2
Bm = 2.0*10^-6; % Damping Coeffecient in Nms
A = [-Ra/La -Kt/La; Kt/Jm -Bm/Jm];
B = [1/La; 0];
Ts = 0;
C = [0  1];
D = 0;
DC_Motor = ss(A,B,C,D,Ts,...
   'InputName','V','OutputName','Y','StateName',{'i','theta_dot'}) %
 State Space Model

Amplifier = tf(24, [1/2500 1]) % Voltage Amplifier with Steady state
 Voltage Gain and Band Width
Plant = DC_Motor * Amplifier % Overall Plant Model
Plant.StateName{3} = 'x3'
[V,E] = eig(A) % Eigen Vectors and Eigen Values of A
Poles = pole(DC_Motor) % Stability Analysis.

step(Plant) % Step Response for a unit step voltage reprsenting the
 rise time, the settling time and the steady state

Controlability  = ctrb(DC_Motor) % Controlability Analysis
Rank_C = rank(Controlability) % Linearly Independent Columns to Prove
 Controlability

Observability = obsv(DC_Motor) % Observability Analysis
Rank_O = rank(Observability) % Linearly Independent Rows to prove
 Observability

% Full State feedback Control
Plant.C = eye(3) % Modifying C Matrix into Identity Matrix
Plant.StateName{3} = 'x3'
Plant.OutputName = {'i','theta_dot','x3'}

%Pole Placement
%K = place(Plant.A, Plant.B, [ -7 + 5.25j, -7 - 5.25j, -21]) % State
 Space Matrices A and B and Desired Pole Position

% Closed Loop State Space Model using Feedback Command
%C1 = feedback(Plant, K)

% step(C1(1))

% Kalman Filter
Q = 10^-3; % Process Covariance Matrix
```

```
R = 10^-4; % Measurement Covariance Matrix
[kalmanf,L,P] = kalman(Plant,Q,R);
kalmanf = kalmanf(1,:) % Output Estimate
sys = parallel(Plant,kalmanf,1,1,[],[]); % Connecting Plant Model and
 Kalman Filter in Parallel

SimModel = feedback(sys,1,2,3);
SimModel = SimModel([1 2],[1 2 3]);
SimModel.InputName
SimModel.OutputName

%this is the part i am trying to comprehend to plot
t = [0:100]'
u = sin(t/6)

rng(10,'twister');
w = sqrt(Q)*randn(length(t),1)
v = sqrt(R)*randn(length(t),1)


DC_Motor =

  A =
                     i  theta_dot
   i             -8807     -53.01
   theta_dot   4.19e+04     -1.905

  B =
                V
   i         1205
   theta_dot    0

  C =
             i  theta_dot
   Y         0          1

  D =
       V
   Y   0

Continuous-time state-space model.


Amplifier =

       24
  ------------
  0.0004 s + 1

Continuous-time transfer function.


Plant =
```

```
  A =
                        i  theta_dot          ?
   i               -8807     -53.01  2.824e+05
   theta_dot    4.19e+04     -1.905          0
   ?                   0          0      -2500

  B =
              u1
   i           0
   theta_dot   0
   ?         256

  C =
              i  theta_dot          ?
   Y          0          1          0

  D =
       u1
   Y    0

Continuous-time state-space model.


Plant =

  A =
                        i  theta_dot         x3
   i               -8807     -53.01  2.824e+05
   theta_dot    4.19e+04     -1.905          0
   x3                  0          0      -2500

  B =
              u1
   i           0
   theta_dot   0
   x3        256

  C =
              i  theta_dot         x3
   Y          0          1          0

  D =
       u1
   Y    0

Continuous-time state-space model.


V =

   -0.1998    0.0062
    0.9798   -1.0000
```

```
E =

   1.0e+03 *

   -8.5473         0
         0   -0.2619


Poles =

   1.0e+03 *

   -8.5473
   -0.2619


Controlability =

   1.0e+07 *

    0.0001   -1.0611
         0    5.0488


Rank_C =

     2


Observability =

   1.0e+04 *

         0    0.0001
    4.1905   -0.0002


Rank_O =

     2


Plant =

  A =
                       i   theta_dot          x3
   i               -8807      -53.01   2.824e+05
   theta_dot    4.19e+04      -1.905           0
   x3                  0           0       -2500

  B =
                u1
   i             0
   theta_dot     0
```

```
   x3          256

  C =
                 i  theta_dot          x3
   Y(1)          1          0           0
   Y(2)          0          1           0
   Y(3)          0          0           1

  D =
         u1
   Y(1)   0
   Y(2)   0
   Y(3)   0

Continuous-time state-space model.


Plant =

  A =
                      i  theta_dot          x3
   i              -8807     -53.01   2.824e+05
   theta_dot   4.19e+04     -1.905           0
   x3                 0          0       -2500

  B =
               u1
   i            0
   theta_dot    0
   x3         256

  C =
                 i  theta_dot          x3
   Y(1)          1          0           0
   Y(2)          0          1           0
   Y(3)          0          0           1

  D =
         u1
   Y(1)   0
   Y(2)   0
   Y(3)   0

Continuous-time state-space model.


Plant =

  A =
                      i  theta_dot          x3
   i              -8807     -53.01   2.824e+05
   theta_dot   4.19e+04     -1.905           0
   x3                 0          0       -2500
```

```
  B = 
             u1
   i          0
   theta_dot  0
   x3       256

  C = 
                    i  theta_dot         x3
   i                1          0          0
   theta_dot        0          1          0
   x3               0          0          1

  D = 
            u1
   i         0
   theta_dot 0
   x3        0

Continuous-time state-space model.


kalmanf =

  A = 
                      i_e  theta_dot_e         x3_e
   i_e         -1.525e+04        -9706    2.819e+05
   theta_dot_e  3.225e+04   -2.675e+04       -485.9
   x3_e            -441.3       -485.9        -2545

  B = 
                      i  theta_dot         x3
   i_e             6438       9653      441.3
   theta_dot_e     9653  2.675e+04      485.9
   x3_e           441.3      485.9       44.5

  C = 
              i_e  theta_dot_e         x3_e
   i_e          1            0            0

  D = 
              i  theta_dot         x3
   i_e        0          0          0

Input groups:        
       Name        Channels
    Measurement     1,2,3  

Output groups:          
         Name          Channels
    OutputEstimate       1     

Continuous-time state-space model.
```

```
ans =

  3×1 cell array

    {'i'        }
    {'theta_dot'}
    {'x3'       }


ans =

  2×1 cell array

    {'i'        }
    {'theta_dot'}


t =

     0
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
    28
    29
    30
    31
    32
    33
    34
```

```
    89
    90
    91
    92
    93
    94
    95
    96
    97
    98
    99
   100


u =

         0
    0.1659
    0.3272
    0.4794
    0.6184
    0.7402
    0.8415
    0.9194
    0.9719
    0.9975
    0.9954
    0.9657
    0.9093
    0.8277
    0.7231
    0.5985
    0.4573
    0.3034
    0.1411
   -0.0251
   -0.1906
   -0.3508
   -0.5013
   -0.6379
   -0.7568
   -0.8548
   -0.9290
   -0.9775
   -0.9990
   -0.9927
   -0.9589
   -0.8986
   -0.8133
   -0.7055
   -0.5782
   -0.4348
   -0.2794
   -0.1163
```

```
 0.0501
 0.2151
 0.3742
 0.5228
 0.6570
 0.7730
 0.8675
 0.9380
 0.9825
 0.9998
 0.9894
 0.9515
 0.8873
 0.7985
 0.6876
 0.5576
 0.4121
 0.2553
 0.0913
-0.0752
-0.2395
-0.3973
-0.5440
-0.6757
-0.7886
-0.8797
-0.9464
-0.9869
-1.0000
-0.9854
-0.9435
-0.8755
-0.7831
-0.6691
-0.5366
-0.3891
-0.2309
-0.0663
 0.1001
 0.2638
 0.4202
 0.5649
 0.6940
 0.8038
 0.8913
 0.9542
 0.9906
 0.9996
 0.9808
 0.9349
 0.8631
 0.7673
 0.6503
 0.5152
```

```
    0.3659
    0.2065
    0.0413
   -0.1250
   -0.2879
   -0.4428
   -0.5854
   -0.7118
   -0.8184


w =

    0.0204
   -0.0463
    0.0122
    0.0354
   -0.0002
   -0.0081
   -0.0370
    0.0185
   -0.0236
   -0.0232
    0.0221
    0.0634
   -0.0395
    0.0019
    0.0197
    0.0035
    0.0338
   -0.0261
    0.0426
    0.0115
    0.0037
   -0.0476
   -0.0135
    0.0106
   -0.0082
   -0.0032
    0.0087
    0.0005
    0.0059
    0.0132
    0.0072
    0.0024
    0.0322
   -0.0188
   -0.0324
   -0.0175
   -0.0168
    0.0483
   -0.0411
    0.0066
    0.0060
```

```
 0.0174
-0.0395
 0.0507
-0.0114
 0.0275
-0.0309
 0.0783
-0.0163
-0.0511
-0.0080
-0.0217
 0.0464
-0.0404
-0.0125
 0.0401
 0.0399
-0.0047
 0.0184
-0.0147
 0.0084
 0.0456
 0.0029
 0.0012
-0.0075
-0.0186
-0.0132
-0.0196
 0.0179
-0.0292
-0.0060
-0.0123
 0.0069
-0.0164
-0.0308
 0.0394
 0.0264
 0.0306
-0.0101
 0.0148
 0.0007
 0.0107
-0.0175
-0.0195
-0.0159
-0.0156
-0.0035
 0.0055
-0.0032
-0.0113
-0.0371
 0.0717
-0.0290
-0.0005
 0.0236
```

```
   -0.0047
   -0.0260
    0.0019
    0.0036
    0.0520
   -0.0344


v =

   -0.0012
    0.0045
    0.0071
    0.0034
   -0.0219
   -0.0026
    0.0010
   -0.0061
    0.0001
    0.0059
   -0.0069
   -0.0090
   -0.0075
   -0.0050
   -0.0099
    0.0085
   -0.0033
   -0.0261
    0.0053
   -0.0012
    0.0058
   -0.0155
   -0.0120
    0.0030
   -0.0142
   -0.0142
    0.0016
   -0.0138
   -0.0043
    0.0108
    0.0055
    0.0014
    0.0006
   -0.0196
    0.0122
    0.0024
    0.0055
    0.0007
    0.0167
    0.0037
    0.0041
   -0.0093
   -0.0029
    0.0038
```

```
-0.0110
-0.0013
-0.0122
 0.0003
 0.0144
-0.0084
 0.0176
 0.0015
 0.0042
-0.0050
-0.0017
-0.0076
-0.0250
-0.0154
-0.0017
 0.0034
 0.0013
-0.0030
-0.0029
 0.0183
 0.0020
 0.0095
 0.0117
 0.0022
 0.0085
 0.0038
-0.0229
 0.0147
-0.0060
-0.0003
 0.0001
-0.0003
 0.0173
-0.0138
-0.0183
 0.0086
 0.0067
-0.0014
-0.0004
-0.0066
-0.0109
 0.0057
-0.0193
-0.0121
 0.0014
-0.0084
 0.0053
 0.0161
-0.0041
-0.0061
 0.0156
 0.0110
-0.0030
 0.0086
```

```
-0.0022
-0.0129
 0.0076
```

Published with MATLAB® R2019a

```
% PROJECT CODE 2
clear
clc
V = 24;
Bm = 2*(10^-6);
L = 8.3*(10^-4);
R = 7.31;
J = 1.05*(10^-6);
Kt = 0.044;

A = [-R/L -Kt/L; Kt/J -Bm/J]
B = [1/L;0]
C = [0 1];
D =0;
p=ss(A,B,C,D);
p_order = order(p)
p_rank = rank(ctrb(A,B))
s = tf('s');
p_motor = Kt/((J*s+Bm)*(L*s+R)+Kt^2);
zpk(p_motor)
% To calculate the state transition matrix for A
pd=c2d(p,0.001)
% Process noise covariance
nQ = 1e-3;
% Measurement noise covariance
nR = 1e-4;
Ts= 0.001;


A =

   1.0e+04 *

   -0.8807   -0.0053
    4.1905   -0.0002


B =

   1.0e+03 *

    1.2048
         0


p_order =

     2


p_rank =
```

```
     2


ans =

      5.0488e+07
  ------------------
  (s+8547) (s+261.9)

Continuous-time zero/pole/gain model.


pd =

  A =
                x1          x2
   x1   -0.02395  -0.004923
   x2      3.891     0.7938

  B =
             u1
   x1  0.1121
   x2   4.648

  C =
       x1  x2
   y1   0   1

  D =
       u1
   y1   0

Sample time: 0.001 seconds
Discrete-time state-space model.
```

Published with MATLAB® R2019a