Control of a Stewart Platform Using Linear Controllers and Extended Kalman Filters

Taylor Apgar, Francis James, Mukhtar Maulimov and Robin Dietrich

Abstract

Due to their high precision, Stewart platforms can be used to control the position and orientation of a workpiece during robotic deburring. In this work, we look at the control of a Stewart platform in the presence of external forces which arise due to interactions between the workpiece and the deburring spindle. A controller based on linearization about a desired operating point is implemented. Feedback consists of leg lengths provided by encoders on the six actuators and a highly accurate kinematic model providing the point of application of force. An Extended Kalman Filter is used to estimate the applied forces and a feed forward term is added to cancel its effects.

1 Introduction

1.1 Stewart Platform

A Stewart platfrom is a 6-Degree Of Freedom (DoF) parallel robot consisting of linear actuators attached in pairs to a top mobile platform and a stationary base. The mobile platform is attached to the actuators via a spherical joint and to the base by a universal joint. The universal joint constrains rotational motion along the axis of the linear motor about the base.

Stewart platforms and other parallel mechanisms are often used in applications that require high precision or require the lifting of heavy loads. An example of a Stewart Platform is shown in Fig. 1. A screenshot of the simulated platform is shown in Fig. 2.

1.2 Application in Deburring

Robotic deburring is the process of removing undesirable projections from the surface of a metal workpiece commonly formed during a milling or casting process. These projections are often just a few microns in size. The unpredictability in the formation of burrs in terms of size and location makes it impractical to eliminate them in the production stage. Consequently, these workpieces need to undergo a deburring operation which is done manually, at least in the aerospace industry, due to the high precision requirements and the inability of current industrial robots to perform hybrid force-position control tasks with the required degree of precision.

We are currently developing a robot capable of meeting these requirements. An integral part of the process is to have a highly precise mechanism capable of maintaining a desired position and orientation while responding to applied forces. To do this, we can use a Stewart platform actuated by backdrivable linear actuators. While this platform itself is mounted on a mobile stage that can coarsely position the workpiece mounted on the platform during deburring, the dynamics arising due to this motion are neglected in this work.

2 Modeling the Stewart Platform

The modeling presented in this work is based on the dynamic modeling of a Stewart platform with piezoelectric actuators [2]. Before getting into the dynamic model of the moving platform, we need to first consider the relationship between the moving platform and the base which is considered as a fixed reference frame. For this, we will need a rotational matrix that will make possible to write the position and orientation of the moving platform with respect to fixed reference frame using a set of X-Y-Z fixed angles (α, β, γ). That is, the moving frame first rotated about the X-axis of the fixed frame with an angle α, then rotated about the Y-axis of the fixed frame with an angle β and finally rotated about the Z-axis of the fixed frame with an angle γ.

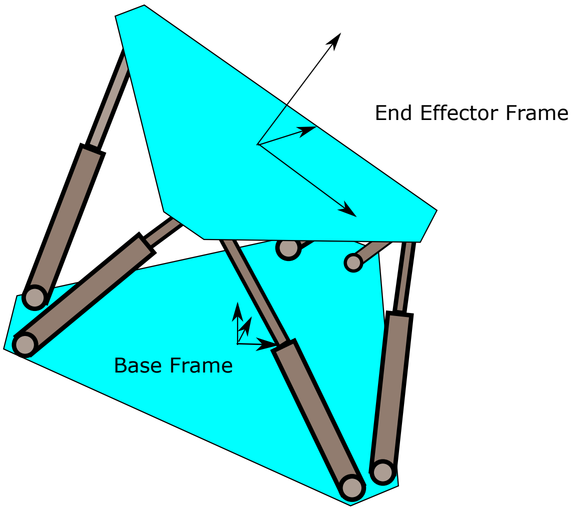

Figure 1: Stewart Platform

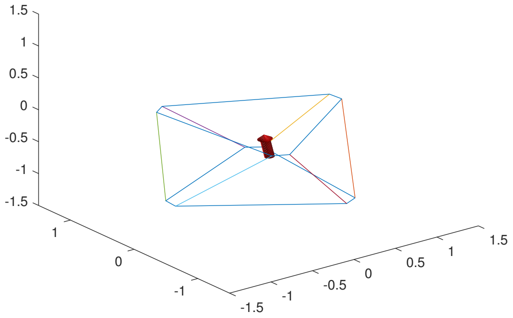

Figure 2: Simulation: The arrow indicates an external force

$$R = \begin{bmatrix} c_\alpha c_\beta & c_\gamma s_\beta s_\alpha - s_\gamma c_\alpha & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\gamma c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\gamma s_\beta c_\alpha - c_\gamma s_\alpha \\ -s_\beta & c_\beta s_\alpha & c_\beta c_\alpha \end{bmatrix} \tag{1}$$

where c and s represent cosine and sine, respectively. Since $\dot{R}R^T$ has the following characteristics

$$\dot{R}R^T = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{2}$$

$\dot{R}$ can be written as follows:

$$\dot{R} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

2.1 Kinematic Model

Let $P_i^p = [P_{ix}^p \quad P_{iy}^p \quad P_{iz}^p]^T$ be the position vector of the i^{th} spherical joint in the moving platform with respect to moving frame fixed on the moving platform, and $L_i = [L_{ix} \quad L_{iy} \quad L_{iz}]^T$ be the position vector of the i^{th} universal joint in the base plate with respect to fixed frame, $i = 1, 2, ..., 6$, then we can represent the absolute coordinate vector of the i^{th} spherical joint with respect to fixed (base) frame as

$$P_i^B = T + RP_i^p \tag{3}$$

where R and T are the rotational matrix and the position vector of origin of moving frame with respect to base fixed frame, respectively.

We will treat the legs of the Steward Platform as lines and represent them as line coordinates. Let S_i be the line vector of the i^{th} from L_i to P_i^p, or

$$S_i = P_i^B - L_i \tag{4}$$

then the six component vector of Plücker coordinates of the i^{th} line will be represented as

$$B_i = [s_i^T M_i^T]^T = [s_{ix} \quad s_{iy} \quad s_{iz} \quad M_{ix} \quad M_{iy} \quad M_{iz}]^T \tag{5}$$

where

$$s_i = \frac{S_i}{|S_i|}$$

$$M_i = RP_i^p \times s_i$$

where s_i is the unit vector along the i^{th} leg from the base to the moving platform, and M_i is a transformation moment vector that gives the moment due to a unit force applied at P_i^p along the i^{th} leg about the origin of the moving frame. Then, the Plücker coordinate matrix of the Steward Platform can be written as

$$B = [B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5 \quad B_6] = \begin{bmatrix} s_{1x} & s_{2x} & s_{3x} & s_{4x} & s_{5x} & s_{6x} \\ s_{1y} & s_{2y} & s_{3y} & s_{4y} & s_{5y} & s_{6y} \\ s_{1z} & s_{2z} & s_{3z} & s_{4z} & s_{5z} & s_{6z} \\ M_{1x} & M_{2x} & M_{3x} & M_{4x} & M_{5x} & M_{6x} \\ M_{1y} & M_{2y} & M_{3y} & M_{4y} & M_{5y} & M_{6y} \\ M_{1z} & M_{2z} & M_{3z} & M_{4z} & M_{5z} & M_{6z} \end{bmatrix} = \begin{bmatrix} s_{3\times6} \\ M_{3\times6} \end{bmatrix} \tag{6}$$

2.2 Dynamic Model

From the free body diagram of the moving platform, we can write the translation and rotation motion equations of the moving platform as follows:

$$F_I + F_G + F_E + \sum_{i=1}^{6} F_i = 0 \tag{7}$$

$$\tau_I + \tau_E + \sum_{i=1}^{6} \tau_i = 0 \tag{8}$$

where F_I and τ_I are the inertia force and moment of the moving platform, F_i is the force vector acting on the moving platform at the spherical joint by each leg with respect to base, $i = 1, 2, ..., 6$, F_G is the gravitational force, F_E and τ_E are the resultant external force and moment acting on the moving platform, and τ_i is the moment vector of F_i about the center of the mass of the moving platform.

F_I and τ_I can be expressed as

$$-F_I = \frac{d}{dt}(mv) = m\dot{v} \tag{9}$$

$$-\tau_i = \frac{d}{dt}(I'w) = \frac{d}{dt}(RIR^T w) = I'\dot{w} + \dot{R}R^T I'w \tag{10}$$

where m and v are the mass and velocity of the moving platform, I is the inertia matrix of the moving platform with respect to moving frame, $I' = RIR^T$ is the inertia matrix of the moving platform about an inertial frame whose origin coincides with moving frame of the platform and with the axes parallel to fixed base platform, and w is the instantaneous rotational velocity vector of the moving platform.

Knowing $u = [u_1, u_2, ..., u_6]^T$, where u_i is the magnitude of the axial force acting on the i^{th} leg as an input to the system, we need transformation matrix, B, that will give us force and moment vectors F_i and τ_i acting on each leg of the moving platform. The detailed derivation of B matrix is given in 6.

Then, writing equations 9 and 10, in a more convenient form, we will get

$$\begin{bmatrix} m(\dot{v}-g)-F_E \\ I'\dot{w}+\dot{R}R^T I'w-\tau_E \end{bmatrix} = Bu \tag{11}$$

where $g = [0 \quad 0 \quad -g_c]^T$ is the gravitational acceleration vector, and g_c is the gravitational constant. Equation 11 can be written in vector-matrix form as follows:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q,f_e) = B(q)u \tag{12}$$

where

$$q = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$$

$$H(q) = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I'_{11} & I'_{12} & I'_{13} \\ 0 & 0 & 0 & I'_{21} & I'_{22} & I'_{23} \\ 0 & 0 & 0 & I'_{31} & I'_{32} & I'_{33} \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{45} & c_{46} \\ 0 & 0 & 0 & c_{54} & 0 & c_{56} \\ 0 & 0 & 0 & c_{64} & c_{65} & 0 \end{bmatrix}$$

$$c_{45} = I'_{31}w_x + I'_{32}w_y + I'_{33}w_z$$

$$c_{46} = -(I'_{21}w_x + I'_{22}w_y + I'_{23}w_z)$$

$$c_{54} = c_{46}$$

$$c_{56} = I'_{11}w_x + I'_{12}w_y + I'_{13}w_z$$

$$c_{64} = -c_{46}$$

$$c_{65} = -c_{56}$$

$$G(q,f_e) = [-F_{Ex} \quad -F_{Ey} \quad mg_c - F_{Ez} \quad -\tau_{Ex} \quad -\tau_{Ey} \quad -\tau_{Ez}]^T$$

where $w_x = \dot{\theta}_x$, $w_y = \dot{\theta}_y$ and $w_z = \dot{\theta}_z$, and $H(q)$ is the mass matrix, $C(q,\dot{q})$ is the Coriolis matrix, $G(q,f_e)$ is the external force matrix, and $B(q)$ is input matrix.

2.3 States

The Stewart platform used in this work is shown in Fig. [1]. The End Effector or platform frame is fixed in relation to the moving platform. The position of the platform with respect to the base coordinate frame is denoted by $[t_x, t_y, t_z]^T$. The orientation of this platform frame is defined by fixed angular rotations about the X-axis, the Y-axis and the Z-axis in the sequence X-Y-Z. These rotations are denoted by $[\theta_x, \theta_y, \theta_z]^T$. The instantaneous velocities are represented by $[\dot{t}_x, \dot{t}_y, \dot{t}_z]^T$ and the instantaneous rotational velocities about the X, Y and Z axes of the base frame are denoted

by $[\omega_x, \omega_y, \omega_z]^T$ respectively.
Defining the vectors q and $\dot{q}$ are given by Eq. 13

$$q = \begin{bmatrix} t_x \\ t_y \\ t_z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \text{ and } \dot{q} = \begin{bmatrix} \dot{t}_x \\ \dot{t}_y \\ \dot{t}_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{13}$$

The state variable used for writing the state space equation is χ and is given by Eq. 14.

$$\chi = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{14}$$

2.4 Control Inputs

The control inputs to the system are the forces applied to the platform by each actuator. This is denoted by the vector u in Eq. (12).

2.5 Measurements

For obtaining state feedback, each leg has an attached encoder that gives a slightly noisy estimate of its length. Based on the forward kinematic model, an estimate of the platform position and orientation can be obtained.
For robotic deburring, the deburring spindle is fixed and can only apply force at that position although the direction and magnitude of the applied force can vary depending on operating conditions such as orientation of the platform, depth of cut, spindle speed and cutting speed among others. In this work, we assume that we have direct access to the location where this deburring force acts on the platform relative to the origin of the platform coordinates. For real world applications, the knowledge of the platform position and orientation, base position and spindle position can be used to compute this point of application of force.

2.6 System Architecture

Figure 3 shows the design of the Stewart platform system. Starting in the upper left is the controller which is trying to maintain a position and velocity of zero for each of the 6 axes. It does this by using the feedback controller (seen in gain block K) which is 12x12 dimensional matrix. A feedforward term is also added which is an estimate of the external force acting on the platform.

This control action u is then applied to each of the actuators in the real system which is simulated by the plant. The plant diagram is shown in the upper right part of the system architecture diagram. This takes the control input u, adds the external force disturbance to it, then applies it to the nonlinear dynamics simulator. The output of the nonlinear dynamics block is the state derivatives which is integrated into the new state. The new state then goes to the nonlinear observation block which simulates what the measurements would be given the current state. Next Gaussian noise is added to each of the measurements which is then input into the observer. The observer is an extended Kalman filter which estimates the 12 control states as well as an additional 6 states which are used to calculate the feedforward forces. The EKF will be discussed in more detail in section 5.

3 Plant

The plant simulates both the nonlinear dynamics and the nonlinear measurements. The best way to visualize the nonlinear dynamics is by using standard manipulator form. The equations of motion in standard manipulator form can be seen in equation 15.

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q, f_e) = B(q)u \tag{15}$$

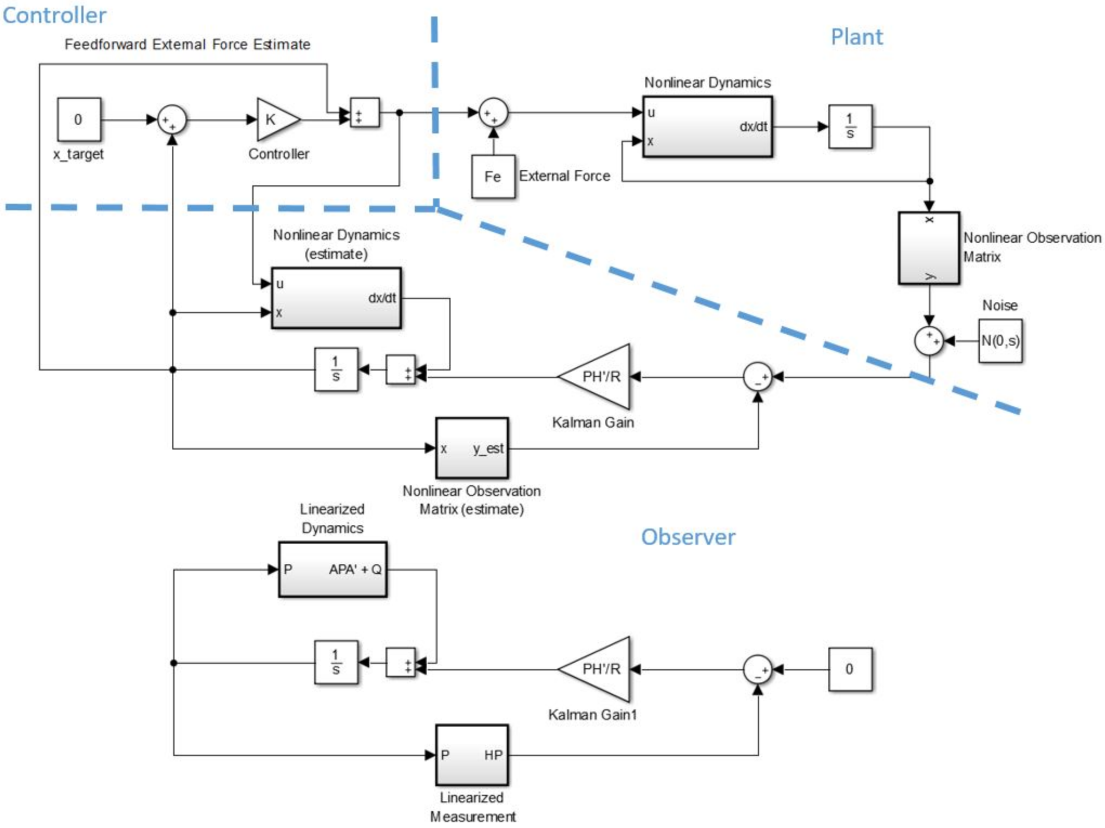

Figure 3: System Architecture

where $H(q)$ is the mass matrix, $C(q, \dot{q})$ is the Coriolis matrix, $G(q, f_e)$ contains the constant force terms, and $B(q)$ maps the control input vector u to the state. The nonlinear dynamics update then takes the form seen in equation 16

$$\dot{\chi} = f(\chi, u) = \begin{bmatrix} \dot{q} \\ H^{-1}(q)[B(q)u - C(q, \dot{q})\dot{q} - G(q, f_e)] \end{bmatrix} \tag{16}$$

As previously mentioned the relationship between the measurements and the states is also nonlinear. There are 9 measurements total, 1 measurement for each of the 6 encoders, and then a measurement of the 3 dimensional pose where the force is applied. The addition of external force estimates into the state is discussed in much more detail in section 5. The expected encoder measurements can be calculated from the current state by using equation 17.

$$y_{enc} = |T(q) + R(q)P_p - B_b| \tag{17}$$

where $T(q) + R(q)P_p$ maps the link joint offset P_p from the platform frame to the base frame and B_b is the other link joint in the base frame. Equation 17 is simply the magnitude of the link vector. The external force positions are added to observers state so converting from state to measurement is simply the identity matrix. The total measurement vector is then calculated using equation 18.

$$y = h(\chi) = \begin{bmatrix} |T(q) + R(q)P_p - B_b| \\ \chi[16:18] \end{bmatrix} \tag{18}$$

4 Controller

4.1 Linearized Equation

For control of this platform, a linear feedback controller based on linearization of the model about the desired point is used. The dynamic equation in Eq. 12 is rewritten as shown in Eq. 19 and then linearzied as shown in Eq. 20

$$\ddot{q} = H^{-1}(q)[B(q)u - C(q, \dot{q})\dot{q} - G(q)] = f(q, \dot{q}, u) \tag{19}$$

$$\ddot{q} \approx \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} \cdots & \frac{\partial f_1}{\partial \dot{q}_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} \cdots & \frac{\partial f_2}{\partial \dot{q}_n} \\ \vdots & \vdots \ldots & \vdots \\ \frac{\partial f_n}{\partial q_1} & \frac{\partial f_n}{\partial q_2} \cdots & \frac{\partial f_n}{\partial \dot{q}_n} \end{bmatrix}\Bigg|_{y=0} \begin{bmatrix} q_1 - q_1^0 \\ q_2 - q_2^0 \\ \vdots \\ \dot{q}_n - \dot{q}_n^0 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \cdots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial u_2}{\partial q_2} \cdots & \frac{\partial f_2}{\partial u_n} \\ \vdots & \vdots \ldots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix}\Bigg|_{y=0} \begin{bmatrix} u_1 - u_1^0 \\ u_2 - u_2^0 \\ \vdots \\ u_n - u_n^0 \end{bmatrix} \tag{20}$$

Here, $y = 0$ denotes the operating point which is $[q_0, \dot{q}_0, u_0]^T = [0, 0, u_0]^T$ where u_0 is the force required to keep the platform stationary when gravity is the only external force. Using this expression for $\ddot{q}$, the state space equation can be written in the standard linear form.

$$\dot{\chi} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = A\chi + Du \tag{21}$$

4.2 Controllability

The controllability matrix P needs to have rank 12 for controlling all 12 variables used to define the state of the system. At the operating point $\chi = 0$, it turns out that a full rank matrix is formed by choosing P as in Eq. 22

$$P = \left[\begin{array}{c:c} D & AD \end{array} \right] \tag{22}$$

By choosing P of this form, the controllability index for each input $(\mu_1, \mu_2, ..., \mu_6)$ is 2. For designing the controller, the terms in the controllability matrix P are rearranged as in Eq. 23.

$$M = \left[\begin{array}{cc:cc:c:cc} d_1 & Ad_1 & d_2 & Ad_2 & ... & d_6 & Ad_6 \end{array} \right] \tag{23}$$

4.3 Canonical Controllable Form

The inverse of this matrix M is written row-wise with the notation

$$M^{-1} = \left[\begin{array}{c} m_{11} \\ m_{12} \\ \hdashline m_{21} \\ m_{22} \\ \hdashline \vdots \\ \hdashline m_{61} \\ m_{62} \end{array}\right] \tag{24}$$

The transformation matrix T to convert the state space dynamic equation into controllable canonical form is given by

$$T = \left[\begin{array}{c} m_{12} \\ m_{12}A \\ \hdashline m_{22} \\ m_{22}A \\ \hdashline \vdots \\ \hdashline m_{62} \\ m_{62}A \end{array}\right] \tag{25}$$

The transformations $\bar{X} = TX$, $\bar{A} = TAT^{-1}$ and $\bar{D} = TD$ give us an equation in the controllable canonical form

$$\dot{\bar{\chi}} = \bar{A}\bar{\chi} + \bar{D}u \tag{26}$$

Here, $\bar{A}$ and $\bar{D}$ have the multi-input canonical form structure.

4.4 Controller Design

The input u can be chosen as a function of the state according to the control law $u = \bar{K}\bar{\chi}$Eq. This results in the equation 27

$$\dot{\chi} = (\bar{A} + \bar{D}\bar{K})\bar{\chi} \tag{27}$$

$\bar{K}$ is chosen such that the poles of each subsystem in $\bar{A}_{eq}$ are at $(-2, -3)$. These poles were chosen to be small so that the observer settling time is faster than the controller settling time which is required for stability. Note that since the canonical form was obtained after a series of transformations, the setting of the poles of each subsystem may not correspond to something tangible. For the actual controller that operates on the physical state values χ, the matrix $\bar{K}$ is transformed back to the physical coordinate system using the transformation Eq. 28.

$$K = \bar{K}T \tag{28}$$

This K is indicated in the controller part of the system architecture shown in Fig. 3.

5 Observer

5.1 Extended Kalman Filter

The feedback controller employed in this work has a proportional position and a proportional velocity gain for each of the axes. Under external forces this controller will have a steady state error since there is no integral control. In the real world deburring example the system will have some way of measuring where the force is applied. If we know where the force will be applied and we can describe the impact the external force would have on one of the measurable states then we can estimate the magnitude and direction of the external force by using an extended Kalman filter. The state used by the extended Kalman filter is seen in equations 29 and 30.

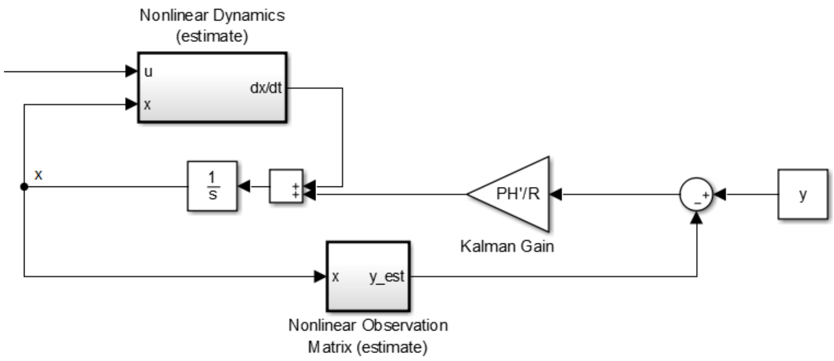

Figure 4: Kalman State Regulator

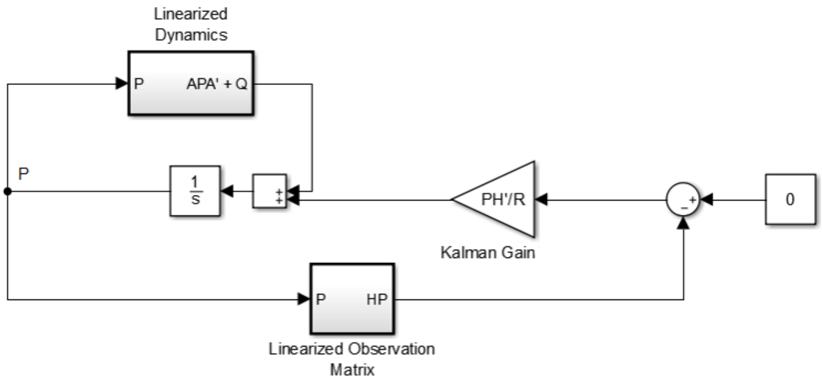

Figure 5: Kalman Covariance Regulator

$$\chi = \begin{bmatrix} q \\ \dot{q} \\ f_e \end{bmatrix} \tag{29}$$

$$f_e = \begin{bmatrix} \vec{f_e} \\ \vec{r_f} \end{bmatrix} \tag{30}$$

where $\vec{f_e}$ is the unobserved 3 dimensional force vector and $\vec{r_e}$ is the observed moment arm where the force is applied. Therefore, the equation for calculating the external forces and torques applied to the platform is equation 31.

$$G(q, f_e) = \begin{bmatrix} \vec{f_e} + m\vec{g} \\ \vec{r_f} \times \vec{f_e} \end{bmatrix} \tag{31}$$

In the EKF there are two main regulators that are running simultaneously. The first is the state regulator which takes a very familiar form and can be seen in Figure 4. This looks very similar to the standard linear observer model. The left side of the diagram uses the control input and previous state to calculate the predicted derivative. The right side of the diagram calculates the difference between the measurement and the measurement predicted by the previous state estimate. It takes this difference and updates the current state estimate according to the Kalman gain. In the typical linear observer this gain would be held constant, however in a Kalman filter this gain is computed according to the covariance matrix P.

The second regulator of the EKF is the covariance regulator which is trying to limit the covariance P as seen in Figure 5. The left side of the diagram is updating the covariance based on the linearized dynamics and model covariance Q. This is known as the apriori or model update. The a posteriori update is on the right side of the diagram. The Kalman gain K is updated according to P and is used to calulate the a posteriori estimate of the covariance matrix. The full EKF algorithm can be seen in Algorithm 1.

The EKF seen in Algorithm 1 is comprised of two sets of equations, the model update equations and the measurement update equations. First, the apriori state update in line 2 uses the previous

Algorithm 1 Extended Kalman Filter

```
Require: $\chi^+(t^-), P^+(t^-)$: previous a posteriori estimate
1: Model Update Equations
2: $\dot{\chi} \leftarrow f(\chi^+(t^-), u)$        ▷ update state according to nonlinear dynamics. Equation 16
3: $P^-(t) = A(t)P^+(t^-)A^T(t) + Q(t)$        ▷ update apriori covariance
4: Measurement Update Equations
5: $K(t) = \frac{P^-(t)H(t)^T}{H(t)P^-(t)H(t)^T + R(t)}$        ▷ calculate Kalman gain
6: $\chi^+(t) = \chi^-(t) + K(t)[y - h(\chi^-(t))]$        ▷ update state according to measurement
7: $P^+(t) = (I - K(t)H(t))P^-(t)$        ▷ update a posteriori covariance
       return $\chi^+(t), P^+(t)$
```

state estimate, action and the nonlinear dynamics from equation 16 to compute an estimate of the current state estimate derivative. The apriori state estimate is then updated by integrating the derivative estimate with the previous state estimate. On line 3 the covariance is updated according to the linearized dynamics A(t) and the constant model covariance Q(t).

In the measurement update equations the apriori estimates are updated according to the new measurements. First the Kalman gain is calulated in line 5. This uses the apriori covariance along with the linearized measurement matrix H(t) and the constant measurement covariance R(t). After a closer look we can see that the Kalman gain is a ratio between the apriori covariance (which is dependent on the model covariance and linearized dynamics) and the constant measurement covariance. If we say we trust our measurements more than our model then the Kalman gain moves closer to 1 whereas if we say we trust our model more than our measurement it moves closer to 0. Line 6 shows the aposteriori state estimate update where $y - h(\chi^-(t))$ is called the measurement residual. The measurement residual is the difference between the measurement and the predicted measurement according to the apriori model update. If we trust our measurements the Kalman gain will increase and then the measurement will have a larger impact on updating the state estimate. The last step in the EKF is in line 7 which updates the covariance. [1]

5.2 Linearization

The Extended Kalman filter is used for system with nonlinear dynamics or nonlinear measurements. An important attribute of the EKF is in the linearization. Two Jacobians that need to be calculated at every time step is the model Jacobian A(t) and the measurement Jacobian H(t). The linearized dynamics equations are shown in equations 32 through 36.

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \\ \dot{f}_e \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ \frac{\partial \ddot{q}}{\partial q} & \frac{\partial \ddot{q}}{\partial \dot{q}} & \frac{\partial \ddot{q}}{\partial f_e} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \\ f_e \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial \ddot{q}}{\partial u} \\ 0 \end{bmatrix} \tag{32}$$

$$\frac{\partial \ddot{q}}{\partial q} = H^{-1}(q)\left(\frac{\partial \ddot{q}}{\partial q} - \frac{\partial C(q,\dot{q})}{\partial q}\dot{q} - \frac{\partial G(q,f_e)}{\partial q}\right) + \frac{\partial H^{-1}(q)}{\partial q}\left(B(q)u - C(q,\dot{q})\dot{q} + G(q,f_e)\right) \tag{33}$$

$$\frac{\partial \ddot{q}}{\partial \dot{q}} = H^{-1}(q)C(q,\dot{q}) \tag{34}$$

$$\frac{\partial \ddot{q}}{\partial f_e} = H^{-1}(q)\frac{\partial G(q,f_e)}{\partial f_e} \tag{35}$$

$$\frac{\partial \ddot{q}}{\partial u} = H^{-1}(q)B(q) \tag{36}$$

Equation 32 takes the familiar A(t)$\chi(t)$ + B(t)u(t) form. Where equations 33 through 35 are used to calculate the elements of the A(t) matrix. Equation 35 is plugged into the A(t) matrix and used to update the covariance matrix $P^-(t)$ on line 3 of algorithm 1. This tells the EKF how the unobserved external forces are related to the observed states. When the Kalman gain is calculated as a function of $P^-(t)$ it uses the measurement residual to update unobserved states.

The nonlinear measurement equations also have to be linearized to calculate matrix H(t). The nonlinear measurement equations are given in equation 18 and the linearized measurement matrix H(t) can be calculated using equation 37.

$$H(t) = \frac{\partial y}{\partial \chi} = \begin{bmatrix} \frac{T(q)+R(q)P_p-B_b}{|T(q)+R(q)P_p-B_b|} & \frac{\partial R(q)}{\partial q}\frac{T(q)+R(q)P_p-B_b}{|T(q)+R(q)P_p-B_b|} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \tag{37}$$

The first row of H(t) is for the encoder measurements and is actually 6 × 18 and the second row of H(t) is for the external force position and is 3 × 18. As previously explained the external force position was added to the observers state so the relationship to the measurement is simply the identity matrix. The first column of H(t) corresponds to the linear displacements, the second column corresponds to the angular displacements, the third column corresponds to the linear and rotational velocities, the fourth column corresponds to the external forces, and the last column corresponds to the external force position.

5.3 Observability

Before considering force estimation there were 12 states, 6 states describing the platforms position in 3 dimensional space, and 6 states describing the velocity of the platform. The minimal set of observations for this system is 6. This is because the platform can move freely in 6 dimensions but the velocity and position of the platform are coupled through the linearized dynamics matrix A(t). However, when we add external force estimation then the number of states increases to 18. The system as described is fully observable as the rank of the observability matrix equals 18. It is interesting to see where the observation contributions come from and important when considering if there are any other states we can estimate through the EKF instead of having to observe them directly.

Table 1: Observability Analysis

	r_f observed	r_f unobserved		1 axis observed	
		$f_e = 0$	f_e != 0	$f_e = 0$	f_e != 0
$H(t)$	9	6	6	7	7
$H(t)A(t)$	6	6	6	6	6
$H(t)A(t)^2$	3	3	5	3	5

Table 1 shows the rank of the observability matrix under a couple different system configurations. The first column shows the system configuration used in this paper where the 3 external force positions are observed directly. This combined with the 6 encoder measurements comprises the rank of 9 for the standalone H(t) matrix. When we add H(t)A(t) we add another 6 due to the velocities whose relationships are carried in the A(t) matrix. Lastly, another 3 are added from the second derivative since the external forces are related directly related to the accelerations.

The next question would be is there a way to start estimating more states so we don't need as many sensors. The second and third columns show the observability rank if we didn't observe any of the 3 external force positions. If no external forces are applied the rank drops to 15. However, if some external force is applied then some of the states become indirectly observable but the rank which is 17 is still below the required observaiblity rank of 18. Interestingly, if we just add in observation of one of the axes where the external force is applied we can get the system back to full observability in the presence of an external force. Since the external force isn't always applied the system would switch between full and partial observability. However, if we extend the extended Kalman filter slightly further to a Multi-Hypothesis Extended Kalman filter we could use a required rank of 12 when we estimate no external force and a required rank of 18 when there is an external force.

6 Results

6.1 Experiments

To test the performance of our controller and observer we conducted several different experiments. First we compared the results with and without the feedforward controller. Then we look at how well the EKF is able to estimate the unobserved states. Then we see the feedforward system response when the external force position is varied. Lastly we look at the system response under substantial Gaussian noise.

6.2 With and Without Feedforward

In the first experiment we started the platform at it's desired position then applied [1, -0.5, 7] N of external force at [0.1, 0, 0] m for 2 seconds. Figure 6 shows the platform positions as the external force is applied and released. As expected there is significant displacement and steady state error since there is no integral term. The max Z displacement is around 0.75 m and the max rotational displacement about the y axis is -0.4 rads. Figure 7 shows the position response with the feedforward force applied. We see that as the external force is being estimated more and more accurately the position error is decreasing. Then when the force stops being applied the EKF needs to readapt its estimate which explains the second peak. The maximum Z displacement is 0.1 and the maximum rotational displacement about the y axis is -0.045. This shows around an 8x improvement from the system without external force estimation.

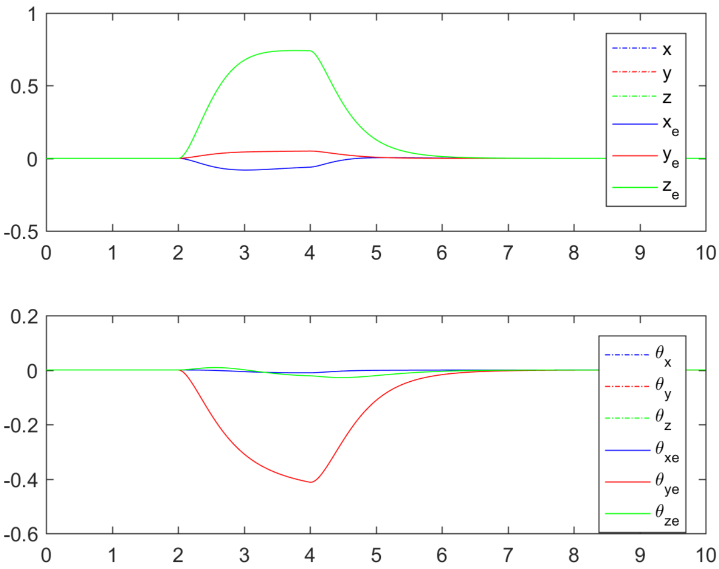

Figure 6: Without Feedforward Position Response

6.3 Estimates of Unobserved Observed States

Next we see how well the EKF is able to estimate the states it can't observe directly. As a reminder the EKF is observing the positions through the encoder and the position where the external force is applied. The EKF is estimating the velocities and the external force vector. The first plot in figure 8 shows the actual and estimated external forces. The forces are applied at 2 seconds and it takes the EKF 0.15 seconds to reach close to the external force estimate. Figure 9 shows the linear and rotational velocities. The peak of the velocity curves correspond to when the EKF force estimates reach the actual external forces being applied since this feedforward causes the platform to stop accelerating. Once the feedforward term kicks in the feedback controller is able to bring the platform back to the target position. This can be seen in the slower velocity response after the initial disturbance.

The other interesting thing of note in these two figures is the estimation error. When the external force is first applied there is an obvious delay in estimation in the EKF but also some of that error gets picked up by the velocity estimate. The EKF is essentially seeing a discrepancy in the accelerations and it tries to determine which estimated states to dump that error into. Once the force estimate reaches the actual then the velocity estimate improves. The controller has a relatively slow response so small errors like this do not have a large impact.

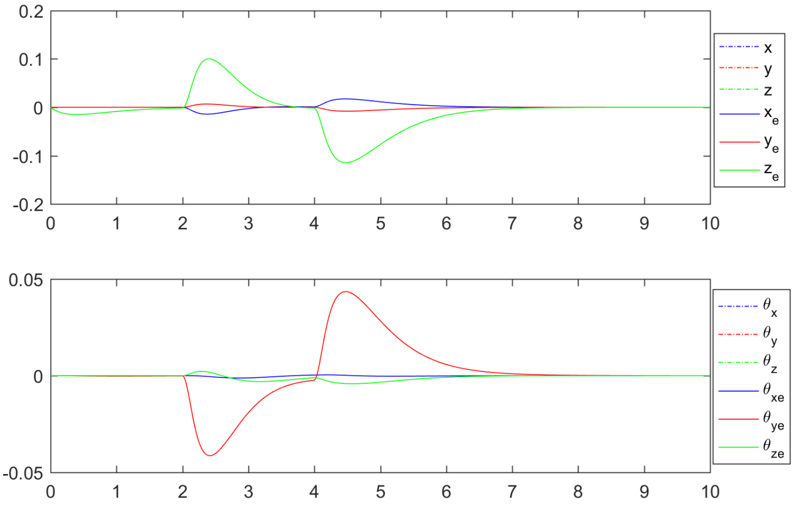

Figure 7: With Feedforward Position Response

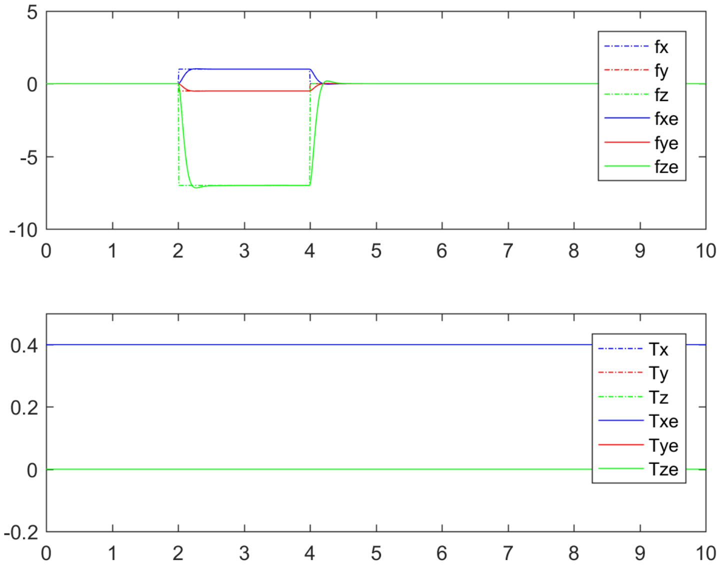

Figure 8: With Feedforward Force Estimates

6.4 Varying Force Location

In the next set of experiments we used the same force vector but move the application position in a 0.4 m radius circle about the center of the platform. Since there is naturally a delay in estimating the unobserved states we want to see how well it is able to track the estimate when it is constantly changing. Figure 10 shows how the force position was varied quickly in a circle about the center of the platform. The main torque being applied to the platform is coming from the force in the Z direction. In figure 11 that the angular displacement never quite settles but constantly varies around 0.005 rads. Although this is a very small error it is still interesting since as we saw in figure 10 the force and the position where the force is applied is being estimated with near perfect accuracy. As we look at figure 12 we see that the angular displacement error is actually due to an error in estimating the rotational velocities. This happens due to a tradeoff we have to make as the designers of the EKF. In order to have more responsive velocity estimates we would have to increase the model covariance for the velocity states, however in doing so we would also decrease

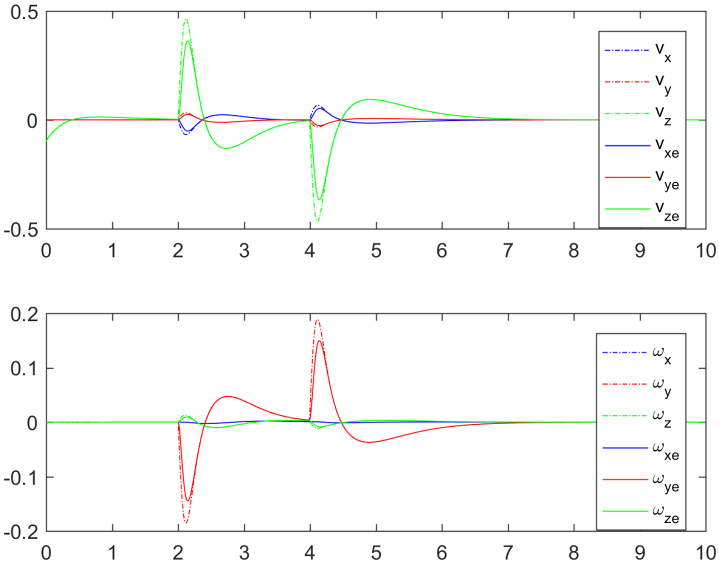

Figure 9: With Feedforward Velocity Estimates

the force estimate response time which actually leads to much larger errors.

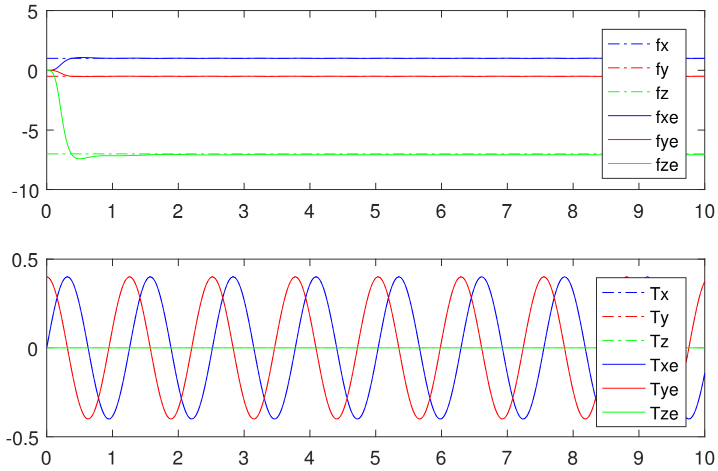

Figure 10: Varying Force Applied to Platform

6.5 Response with Sensor Noise

In the next set of experiments we add a substantial amount of Gaussian noise to the force position measurement. We chose to look at noise specifically for this measurement since in real life encoders have very small amount of noise and it is not Gaussian. For this experiment we used the same force profile as in the previous experiment but we also added zero mean Gaussian noise with a standard deviation of 50 mm to the force position measurement. The second plot in figure 13 shows how substantial the added Gaussian noise is. As expected the estimates are tracking the actual values quite well. The velocity estimates and the position tracking error are slightly worse than in the previous experiment. This is primarily due to the fact that after adding the Gaussian noise we had to change the measurement covariance for the force position measurements. Because all of the states are indirectly related through the dynamics this changes the estimation response

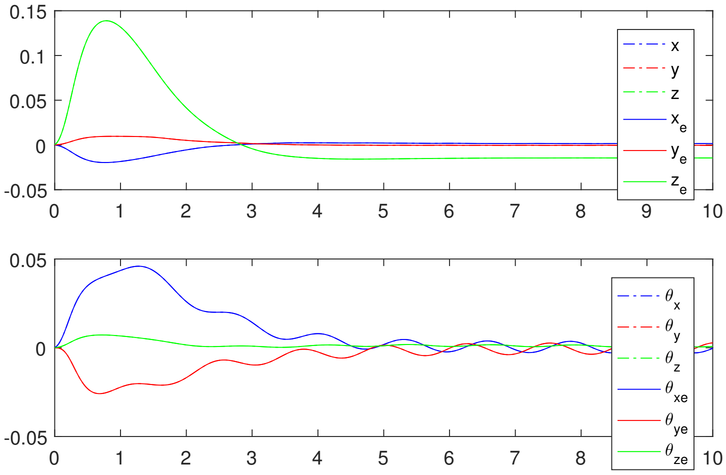

Figure 11: Position Following Error with Varying Torques

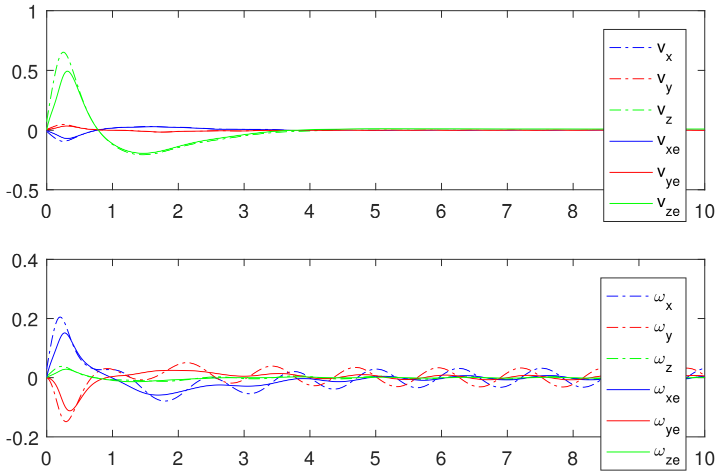

Figure 12: Velocity Estimation Error

of the other states. Of course if this were a real system we would tune the model and measurement covariances to match reality

References

[1] Crassidis, J. L., and Junkins, J. L. Optimal estimation of dynamic systems, 2011.

[2] Lin, L.-C., and Tsay, M.-U. Modeling and control of micropositioning systems using stewart platforms. *Journal of Robotic Systems 17*, 1 (2000), 17–52.

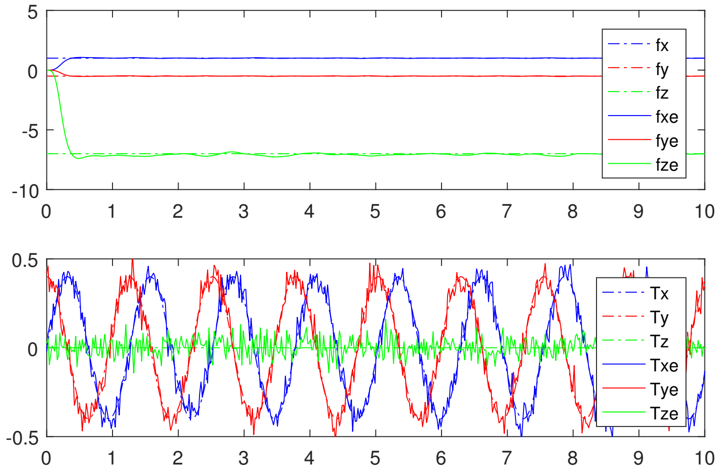

Figure 13: Force Estimates with Noisy Force Position Measurement

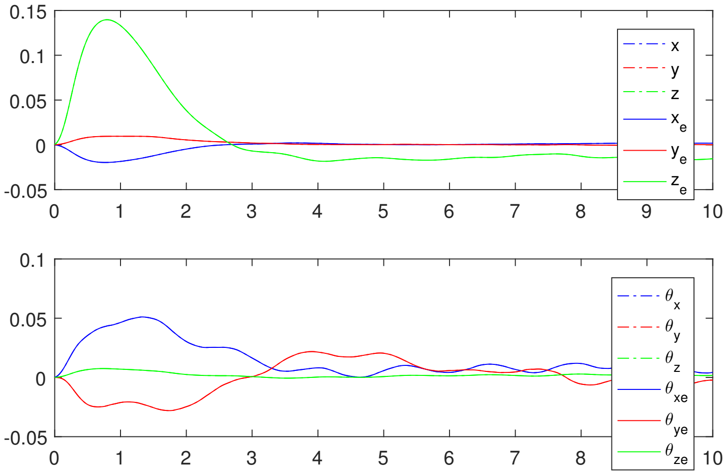

Figure 14: Position Following Error with Noisy Force Position Measurement

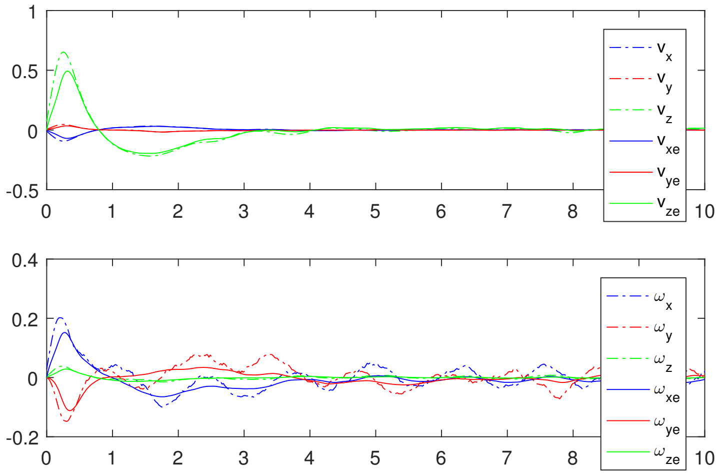

Figure 15: Velocity Estimation Error with Noisy Force Position Measurement