# Table of Contents

Abstract.................................................................................................................. 4
Introduction.............................................................................................................. 5
  Physical Plant Description
  System Connection......................................................................................... 6
System Identification
  Analytical Modeling......................................................................................... 7
    State Space Representation........................................................................... 11
    Transfer Function Form............................................................................... 11
    Plant Decoupling......................................................................................... 12
  Experimental Modeling using Digital Signal Analyzer............................... 14
    Frequency Characterization.......................................................................... 15
    System Isolation............................................................................................ 15
    System Decoupling....................................................................................... 16
    Curve Fitting................................................................................................. 17
Analytical and Experimental Comparison......................................................... 18
Controller Designs................................................................................................ 20
  Methodology.................................................................................................... 20
    Internal Model Principle.............................................................................. 21
    Robust Stability Analysis Framework......................................................... 22
    Selection of Sampling Time......................................................................... 26
  Direct and Indirect Lead-Lag Controller....................................................... 27
    Design
      Continuous-time......................................................................................... 27
      Discrete-time.............................................................................................. 32
    Sensitivity and Complementary Sensitivity Analysis................................. 34
    Robust Stability Analysis............................................................................. 35
    Simulation...................................................................................................... 36
    Implementation.............................................................................................. 39
    Sinusoidal Reference Tracking..................................................................... 41
State Estimation Feedback.................................................................................... 42
  Formulation...................................................................................................... 42
  Design............................................................................................................... 47
  Sensitivity and Complementary Sensitivity Analysis................................. 49
  Robust Stability Analysis............................................................................... 50
  Simulation........................................................................................................ 51
  Implementation................................................................................................. 53
Summary............................................................................................................... 54
ABSTRACT

This report explores various digital controller designs on a magnetic bearing system. Models of the translational and rotational dynamics in the y-direction are obtained analytically and validated against experimental frequency response data. The report illustrates the theory, design, and implementation of several digital controllers with considerations given to stability, robustness, performance, and reference tracking of step and periodic external signals.

The framework of this report starts with an understanding of the magnetic bearing system from a dynamics point of view. With a model of the system in hand, the controllers were motivated and theorized, designed in MATLAB and Simulink environments, and implemented on a xPC setup connected to the MBC500.

From considerations beginning with the classical lead-lag compensator to more modern control designs in repetitive control, the reader should note the improvements, as well as the tradeoffs, as the methodologies progress.

The MATLAB m-files used to conduct analysis and simulations are included in the Appendix. A description of their function appears on the first page in that section.
INTRODUCTION

Physical Plant

The magnetic bearing is a shaft suspended by 4 electromagnetic actuators, two on each end. The actuators are oriented in the horizontal $x$ and vertical directions $y$. There are four Hall Effect sensors placed in a similar manner.

The manufacturer provides an optional analog controller programmed to stabilize the plant. In identifying the plant, these controllers are turned on by loop switches on the left.

Twelve user defined signals exist on the MBC500. For each electromagnet and Hall Effect sensor combination are three signals: input reference $r$, control voltage $u$, and position voltage $y$. Figure X highlights the I/O configuration.

![Figure 1: Inputs and Outputs to the Magnetic Bearing](image)
System Connection

The connection layouts of the coupled and decoupled magnetic bearing system are given in Figures 2 and 3, respectively.

**Figure 2:** Bearing System seen by Controller Cx1 – x and y directions coupled

**Figure 3:** Bearing System seen by Controller Cx1 – x and y directions decoupled
ANALYTICAL MODELING

The modeling of the magnetic bearing is performed separately on the plant and the controller. The signal flow for the plant is

\[ \text{D/A} \rightarrow \text{Voltage} \rightarrow \text{Amplifier} \rightarrow \text{Current} \rightarrow \text{Electromagnets} \rightarrow \text{Force} \rightarrow \text{Mechanical Dynamics} \rightarrow \text{Motion} \rightarrow \text{Sensor Voltage} \rightarrow \text{A/D} \]

And the signal flow for the controller is

\[ \text{A/D} \rightarrow \text{Control} \rightarrow \text{D/A} \]

A mathematical description of the system is next needed to determine its dynamics. The forces and measurements are shown in Figures 1 and 2.

The rigid body dynamics are investigated assuming the shaft does not rotate. This allows for the decoupling of the x and y system states, and their individual input/output descriptions. The table below describes the symbols used and their descriptions. Note that the analysis is exactly the same for the y-direction dynamics.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0$</td>
<td>The horizontal displacement of the center of the rotor’s mass.</td>
</tr>
<tr>
<td>$x_1, x_2$</td>
<td>The horizontal displacements of the rotor at the left and right bearing positions, respectively.</td>
</tr>
<tr>
<td>$X_1, X_2$</td>
<td>The horizontal displacements of the rotor at left and right Hall Effect sensor positions, respectively.</td>
</tr>
<tr>
<td>$\theta$</td>
<td>The angles that the long axis of the rotor makes with the $z$-axis.</td>
</tr>
<tr>
<td>$F_1, F_2$</td>
<td>The forces exerted on the rotor by left and right bearings, respectively.</td>
</tr>
</tbody>
</table>

Table 1: Modeling Definitions

The equations of motions governing the magnetic bearing system are given as

$$\sum F = m\ddot{x}_0 = F_1 + F_2$$
$$\sum M = I_\theta \ddot{\theta} = F_1(\frac{\ell}{2} - l - d) \cos \theta - F_1(\frac{\ell}{2} - l + d) \cos \theta$$

$$x_i = x_0 - (\frac{\ell}{2} - l - d) \sin \theta$$
$$x_2 = x_0 + (\frac{\ell}{2} - l + d) \sin \theta$$
$$X_i = x_0 - (\frac{\ell}{2} - l_i - d) \sin \theta$$
$$X_2 = x_0 + (\frac{\ell}{2} - l_i + d) \sin \theta$$

The set of equations can be compactly represented in state-space form. Applying the small angle approximations $\sin \theta \approx \theta, \cos \theta \approx 1$ to linearize the system, we get

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_0 \\ \dot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ \frac{1}{m} \\ 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(\frac{\ell}{2} - l_i - d) & 0 \\ 1 & 0 & (\frac{\ell}{2} - l_i + d) & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \theta \end{bmatrix}$$

The equation for the magnetic forces can be described by

$$F_i = k \left( \frac{(i_{\text{control}} + 0.5)^2}{(x_i - 0.0004)^2} - k \frac{(i_{\text{control}} - 0.5)^2}{(x_i + 0.0004)^2} \right)$$

Since we assume small displacements in these forces, we can linearize the above equation about its equilibrium point $(x_i, i_{\text{control}}) = (0, 0)$. The Taylor series approximation at this point is
\[ F_t(x_i, i_{\epsilon_{-j}}) = F_t(0,0) + \left[ \frac{\partial F_t}{\partial x_i}(0,0) \right] (x_i - 0) + \left[ \frac{\partial F_t}{\partial i_{\epsilon_{-j}}}(0,0) \right] (i_{\epsilon_{-j}} - 0) \]

with
\[
\frac{\partial F_t}{\partial x_i} = k \left( \frac{-2(i_{\epsilon_{-j}} + 0.5)^2 - 2(i_{\epsilon_{-j}} - 0.5)^2}{(x_i - 0.0004)^3} \right) \quad \Rightarrow \frac{\partial F_t}{\partial x_i} \bigg|_{(0,0)} = -4375
\]
\[
\frac{\partial F_t}{\partial i_{\epsilon_{-j}}} = k \left( \frac{2(i_{\epsilon_{-j}} + 0.5)}{(x_i - 0.0004)^2} - \frac{2(i_{\epsilon_{-j}} - 0.5)}{(x_i + 0.0004)^2} \right) \quad \Rightarrow \frac{\partial F_t}{\partial i_{\epsilon_{-j}}} \bigg|_{(0,0)} = 3.5
\]

Thus, the linearized magnetic force is
\[ F_t = 4375x_i + 3.5i_{\epsilon_{-j}} \]

With the small angle approximation of
\[ x_1 = x_0 - \left( \frac{l}{2} - l - d \right) \sin \theta \approx x_0 - \left( \frac{l}{2} - l - d \right) \theta \]
\[ x_2 = x_0 + \left( \frac{l}{2} - l + d \right) \sin \theta \approx x_0 - \left( \frac{l}{2} - l + d \right) \theta \]

the solution of the magnetic forces is
\[ F_1 = 4375x_0 - 4375\left( \frac{l}{2} - l - d \right) \theta + 3.5i_{\epsilon_{-j}} \]
\[ F_2 = 4375x_0 + 4375\left( \frac{l}{2} - l + d \right) \theta + 3.5i_{\epsilon_{-j}} \]

The state-space representation can now be written as

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} = 
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
i_{\epsilon_{-j}} \\
\theta
\end{bmatrix} + 
\begin{bmatrix}
0 & 0 & 0 & 0 \\
\frac{4375}{8750} & \frac{1}{2} & 0 & 0 \\
-\frac{1}{4(l-d)} & \frac{1}{2} & 0 & 0 \\
\frac{3.5}{8750} & 0 & 0 & 3.5
\end{bmatrix}
\begin{bmatrix}
i_{\epsilon_{-j}} \\
\theta
\end{bmatrix}
\]

\[
\Rightarrow \begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} = 
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\frac{4375}{8750} & \frac{1}{2} & 0 & 0 \\
-\frac{1}{4(l-d)} & \frac{1}{2} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
i_{\epsilon_{-j}} \\
\theta
\end{bmatrix} + 
\begin{bmatrix}
0 & 0 & 0 & 0 \\
\frac{3.5}{8750} & 0 & 0 & 3.5 \\
-\frac{1}{4(l-d)} & \frac{1}{2} & 0 & 0 \\
\frac{1}{4(l+d)} & 0 & 0 & 3.5
\end{bmatrix}
\begin{bmatrix}
i_{\epsilon_{-j}} \\
\theta
\end{bmatrix}
\]
**Current Amplifier Dynamics**

The setup includes a dual-channel current amplifier that is governed by the following differential equation:

\[
\frac{d}{dt}(i_{\text{control}}) = -\frac{1}{2.2 \times 10^{-4}} i_{\text{control}} + \frac{0.25}{2.2 \times 10^{-4}} V_{\text{control}}
\]

**Sensor Dynamics**

The relationship between the sense voltage and the horizontal displacements of the rotor at the Hall Effect sensors is given by

\[
V_{\text{sense}_i} = 5000X_i + (25 \times 10^6)X_i^3
\]

Linearizing with the Taylor approximation at the point \(X_i = 0\), the above equation becomes

\[
V_{\text{sense}}(X_i) = V_{\text{sense}}(0) + \left[ \frac{\partial V_{\text{sense}}}{\partial X_i}(0) \right] (X_i - 0)
\]

With

\[
\frac{\partial V_{\text{sense}}}{\partial X_i} = 5000 + 3(25 \times 10^6)X_i^2 \Rightarrow \frac{\partial V_{\text{sense}}}{\partial X_i}(0) = 5000
\]

the solution becomes

\[
V_{\text{sense}_i} = 5000X_i
\]

The state space representation remains the same with modification to the output equation.

\[
\begin{bmatrix}
V_{\text{sense}_1} \\
V_{\text{sense}_2}
\end{bmatrix} = 5000 \begin{bmatrix}
1 & 0 & \left( \frac{L}{2} - l_e - d \right) & 0 \\
1 & 0 & \left( \frac{L}{2} - l_e + d \right) & 0
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\]
The overall state space representation is

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\frac{d\theta}{dt} \\
\frac{d\phi}{dt} \\
\frac{d\psi}{dt} \\
\frac{d\chi}{dt} \\
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_{1} \\
x_{2} \\
x_{3} \\
x_{4} \\
\end{bmatrix}
\]

where our new state vector is

\[
x_r = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \Theta \\ \dot{\Theta} \\ i_r \\ \dot{i}_r \end{bmatrix}
\]

Transfer Function Representation

The state space model reveals that the system is multiple-input multiple-output (MIMO) in nature. Thus, there exist transfer functions from each input to each output, totaling four in our model. The input-output relationships can be summarized compactly in matrix form as

\[
\begin{bmatrix}
V_{sens1} \\
V_{sens2} \\
\end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} V_{c1} \\
V_{c2} \end{bmatrix}
\]

To write the transfer functions, we need values for the physical constants defined in the model. The system parameters are summarized in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>0.146 meters</td>
<td>Total shaft length</td>
</tr>
<tr>
<td>l</td>
<td>0.024 meters</td>
<td>Distance from magnet to end</td>
</tr>
<tr>
<td>l_2</td>
<td>0.0028 meters</td>
<td>Distance from Hall sensor to end</td>
</tr>
<tr>
<td>I_0</td>
<td>2.53x10^{-4} kg \cdot m^2</td>
<td>Moment of Inertia</td>
</tr>
<tr>
<td>M</td>
<td>0.1427 kg</td>
<td>Mass</td>
</tr>
<tr>
<td>d</td>
<td>~0 meters</td>
<td>Center of Mass Displacement</td>
</tr>
</tbody>
</table>

Table 2: System Parameter Values
With these parameter values, the corresponding state space representation is then

\[
\begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 2
\end{bmatrix}
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4 \\
\dot{x}_5 \\
\dot{x}_6
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
\]

\[
\begin{bmatrix}
y_{V,1} \\
y_{V,2}
\end{bmatrix}
= 
\begin{bmatrix}
5000 & 1 & 0 & -\left(\frac{d}{2} - l_2 - d\right) & 0 & 0 & 0 \\
5000 & 1 & 0 & \left(\frac{d}{2} - l_2 + d\right) & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6
\end{bmatrix}
\]

The Bode plots of the four transfer functions are plotted in Figure 3 using the \textit{ss2tf} command in MATLAB.

![Bode plots of the four transfer functions](image)

\textbf{Figure 3: Frequency Response of Analytical Model}

The figure indicates coupling of translation and rotation since the off-diagonal terms are non-zero. Physically, this means a displacement on one end of the shaft (recorded as a voltage) will send voltages to both sensors. Physically, this makes sense since applying a force to one end will move both ends, with the resulting motion being a composition of translation and rotation. This complicates control design because this is a MIMO system. Finding a way to reduce the system to a SISO one is desirable.

\textit{Plant Decoupling}

We can simplify control design by noting that the translational and rotational dynamics can be decoupled. By defining our inputs appropriately, we can achieve this. Consider applying equal forces in the same direction to both ends of the shaft. The resulting motion is purely translational since the shaft will only displace vertically. By applying the forces on both ends in opposite directions, the shaft will rotate about its center of mass.

Mathematically, we need to apply a transformation the model such that the input and output variables are voltages that affect translation and rotation instead of tip displacement.
where $t$ and $\theta$ denote translation and rotation, respectively. Defining the transformation matrix $T$ as

$$T = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

we get the following:

$$T \begin{bmatrix} V_{\text{sense},t} \\ V_{\text{sense},\theta} \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} T^{-1} \begin{bmatrix} V_{e,1} \\ V_{e,2} \end{bmatrix}$$

$$T \begin{bmatrix} V_{\text{sense},t} \\ V_{\text{sense},\theta} \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} T^{-1} \begin{bmatrix} V_{e,1,1} \\ V_{e,2,0} \end{bmatrix}$$

where our transformed plant is

$$\tilde{G} = T \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} T^{-1} = \begin{bmatrix} \tilde{G}_{11}(s) & \tilde{G}_{12}(s) \\ \tilde{G}_{21}(s) & \tilde{G}_{22}(s) \end{bmatrix}$$

We expect this “new” plant to be decoupled i.e. the off-diagonal elements are zero. Figure X plots the transfer function elements of this matrix. Note that the magnitudes of the off-diagonal transfer functions are much smaller than those on the diagonal. This indicates the system has been decoupled and can perform control design on two SISO systems instead of a MIMO one. The transformation of our state-space that achieves this is

$$\hat{x}_r = T x_r$$

where our state transformation matrix $T$ is

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$
Our new state vector is

\[
\hat{x}_r = \begin{bmatrix}
  x_0 \\
  \dot{x}_0 \\
  i_{c1} + i_{c2} \\
  \Theta \\
  \dot{\Theta} \\
  i_{c1} - i_{c2}
\end{bmatrix}
\]

Intuitively, this makes physical sense since the same current-induced force in the same direction applied to both shaft purely translates the shaft, while in the opposite direction rotates it. The resulting A matrix is block diagonal with the translational transfer function on the (1,1) block and rotational on the (2,2) block.

![Figure 4: Frequency Response of Decoupled Analytical Model](image)

The analytical transfer functions for the translation and rotation models are provided later in the report after we examine the experimental modeling of the plant.

**Experimental Modeling Using Digital Signal Analyzer**

The following series of steps are used to identify the MBC500 magnetic bearing system:

1. Frequency Characterization with the Digital Signal Analyzer (DSA)
2. System Isolation
3. Decoupling
4. Frequency Curve Fit for Decoupled System

Model simulation was used to verify the accuracy of the models.
Frequency Characterization

The coupling between the horizontal and vertical directions is ignored in the characterization. When the characterization is performed between directions, the signal is very small. However, the positions are extremely coupled when taken in the same direction. The 2x2 system must be considered as a whole. For simplicity, the analysis will consider only one direction, and can be repeated for the other.

The control signal $u$ cannot be directly controlled, so we circumvent this fact by finding the transfer function between $r$ and $u$ in addition to $r$ to $y$. Using the DSA, we refer to transfer functions as $T_{ru}$ and $T_{ry}$, respectively. Classical linear feedback control theory tells us that

$$T_{ru} = (I + CG)^{-1}$$
$$T_{ry} = (1 + CG)^{-1}G$$

With all the combinations between sides, eight frequency responses are obtained

<table>
<thead>
<tr>
<th>$T_{r1u1}$ with $T_{r1y1}$</th>
<th>$T_{r1u2}$ with $T_{r1y2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{r2u1}$ with $T_{r2y1}$</td>
<td>$T_{r2u2}$ with $T_{r2y2}$</td>
</tr>
</tbody>
</table>

System Isolation

We know that

$$Y = T_{ry} R$$
$$U = T_{ru} R$$

Combining the two, we get

$$Y = T_{ry}T_{ru}^{-1}U$$

The system plant from input $u$ to output $y$ is then defined as

$$G = [G_{11} \ G_{12}] = [T_{r1y1} \ T_{r1y2}] = [T_{r1u1} \ T_{r1u2}]^{-1}$$

$$G = [G_{21} \ G_{22}] = [T_{r2y1} \ T_{r2y2}] = [T_{r2u1} \ T_{r2u2}]^{-1}$$

The experimental frequency responses for the elements of $G$ are shown in Figures 5
**Decoupling**

Because the system outputs due to rotation and translation inputs are independent of each other, we can simplify our model. We are allowed to decouple the system using a transition matrix $T$

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

The transformation is

$$G_d = TGT^{-1} = TGT$$

with the off-diagonals representing the transfer functions between a translational/rotational input and rotational/translational output. Ideally, these are zero, but in practice they are close to zero and several orders of magnitude smaller as shown in Figures 6.

**Figure 5:** Frequency Response of Physical System

**Figure 6:** Frequency Responses of Decoupled Physical System
Curve Fitting

We use MATLAB’S curve fitting function `invfreq` to find a fit to the experimental frequency response. The weighting functions used was

\[ W_t = (f<20)*1.5 + (f<1000)*.5+1e-9; \]

The obtained fits for the translation and rotation models are plotted in Figure X. The transfer functions are also provided.

![Figure 7: Fit and Experimental Frequency Response, Translation](image1)

\[ G_{y,\text{translation}} = -705.3506 \frac{(s - 1588)(s + 1391)}{(s + 4188)(s + 411)(s - 377.2)} \]

![Figure 8: Fit and Experimental Frequency Response, Rotation](image2)

\[ G_{y,\text{rotation}} = -600.6438 \frac{(s - 2495)(s + 1521)}{(s + 3960)(s + 416.5)(s - 435.3)} \]

The fits were assumed to have three poles to match with our analytical models’ poles. Note the existence of one two open-loop stable poles and one unstable pole. Although the analytical model does not have zeros, we decided to include some in the experimental
model to increase the goodness of fit. We will use the experimental models to design controllers.

Comparison of Analytical Model to Experimental and Curve Fit

Figure 9 and 10 compare the system magnitude and phase of our three methods used to characterize the dynamics. The shapes of the analytical plots compare well with the experimental ones, suggesting that our model fits our physical system reasonably well. The accuracy applies especially well at the frequencies below 500 Hz for both magnitude and phase. However, the gain of 5,000 in the sensor dynamics was modified to shrink the magnitude gap; increasing it to 11,000 provided a better match.

![System ID of G_y_translation](image1)

**Figure 9:** System Identification of $G_{y,\text{translation}}$

![System ID of G_y_rotation](image2)

**Figure 10:** System Identification of $G_{y,\text{rotation}}$
The analytical transfer functions are

\[
G_{ya,\text{translation}}(s) = \frac{6.689 \times 10^8}{(s + 4545)(s + 247.6)(s - 247.6)}
\]

\[
G_{ya,\text{rotation}}(s) = \frac{1.298 \times 10^9}{(s + 4545)(s + 288.2)(s - 288.2)}
\]

Thus, we expect the decoupled system for pure translational and rotational dynamics to have third order transfer functions. Comparing to the curve fit transfer functions, the system order matches with exactly one unstable and two unstable poles. However, the analytical model lacks any zeros while the curve fit suggests there are two. All in all, the model gives a fair amount of confidence that it captures a majority of the plant dynamics.
CONTROLLER DESIGNS

As a class, we will use the transfer functions of the translation and rotational of the plant in the y direction

\[ G_{\text{d,translation}}(s) = \frac{-681.1s^2 + 1.843 \cdot 10^5 s + 1.553 \cdot 10^9}{s^3 + 4075s^2 - 4.024 \cdot 10^4 - 6.537 \cdot 10^8} \]

\[ G_{\text{d,rotation}}(s) = \frac{-589.6s^2 + 5.722 \cdot 10^5 s + 2.454 \cdot 10^9}{s^3 + 4056s^2 - 2.441 \cdot 10^4 - 7.692 \cdot 10^8} \]

instead of the ones obtained in the analytical modeling and curve fit. They are both open loop unstable due to poles in the right half complex plane. It turns out that the transfer functions were very similar in form for the x-direction.

Methodology

Step 1: \((G(s), C(s))\)

For each controller presented in this report, we design based on the low-order models for its simplicity.

Step 2: \((G_{\text{zoh}}(z), C(z))\)

We map the controller, if necessary, and the model of the plant into the z-domain for digital implementation via a zero-order hold function and a specified sampling time.

Step 3: \((G_{\text{actual}}, z_{\text{oh}}(z), C(z))\): Simulation

We expect the designed controller will stabilize the plant. Before implementing the controller, we run simulations on a higher order plant model to reduce the possibility for unstable compensation. For this, we use a 10th order model of the original (coupled) plant that we decouple as necessary to implement the translation and rotation controllers.

Step 4: \((G_{\text{actual}}(s), C(z))\): Implementation

We implement the controllers to the actual system, and compare the experimental results to the simulated ones for verification.
**Internal Model Principle**

In controller design, it is often desired to achieve rejection to external inputs to the system. In other words, we desire asymptotic regulation (zero-steady state). This is achieved by inserting the dynamics of the external signal in the feedback path between the external input signal and the regulated output signal.

This method assumes that the dynamics of the external signal are known with arbitrary initial conditions. Mathematically, we want the signal dynamics to appear in the numerator of the controller compensated system such that when the known input is applied, we have cancellation of those dynamics. This is the Internal Model Principle (IMP).

Consider the control diagram in Figure 1.

**Figure 1**: Typical Block Diagram of Feedback Compensated System

IMP applies to the following transfer functions

\[
\frac{y}{d_o}, \frac{y}{d_i}, \frac{e}{r}, \frac{e}{d_i}, \frac{e}{d_o}
\]

but does not to the following:

\[
\frac{y}{r}, \frac{y}{n}, \frac{u}{r}, \frac{u}{d_i}, \frac{u}{d_o}, \frac{u}{n}
\]

In our exploration of controllers, we will apply IMP to eliminate steady-state error in reference tracking to step and sinusoidal signals. The dynamics of these signals are

- **Step Signals**: \( d(k) = \frac{b}{z-1} \)
- **Sinusoidal Signals**: \( d(k) = \frac{B(z)}{z^2 - 2\cos(\omega_hT_s)z + 1} \)
- **Periodic Signals**: \( d(k) = \frac{B(z)}{z^{np} - 1} \)
Robust Stability Analysis Framework

Typically, we analyze gain and phase margins of compensated systems to determine their proximity to instability. Bode plots of the open-loop gains are often plotted and looked at. Recall that those plots contain information from Nyquist plots. A generic Nyquist plot is shown in Figure 1.

![Figure 1: Stability Margins on Nyquist Plot](image)

If we define a radius $\rho$ around the $-1+0j$ point, the required gain margin for stability can be formulated as

$$\frac{1}{1-\rho} \leq GM \leq \frac{1}{1+\rho}$$

where

$$\rho = \min_{\omega} |L - (-1)| = \min_{\omega} |L + 1|$$

This can also be written as

$$\frac{1}{\rho} = \max_{\omega} \left| \frac{1}{L+1} \right| = \max_{\omega} |S| = \|S\|_\infty$$

where $S$ is the sensitivity transfer function. The phase margin can be computed by using the law of cosines with the side opposite the angle of interest having length $\rho$ and the others having length 1.

$$\theta \geq \cos^{-1} \left( \frac{\rho^2 - 2}{2bc} \right)$$

Thus, we have a relationship between phase and gain margin to the closed-loop infinity norm of the sensitivity function.
Another way to formulate robust stability is to consider the feedback diagram shown in Figure 2.

![Figure 2: Robust Stability Model](image)

We represent the plant transfer function by the nominal model $\hat{G}$ we derived before and some unmodelled dynamics, $\delta$.

$$G = \hat{G}(1 + \delta)$$

where

$$\delta(\omega) = \frac{G - \hat{G}}{\hat{G}}$$

By the Small Gain Theorem, the closed-loop system is stable if

1. feedback ($\hat{G}, C$) is stable and $\delta$ is stable.
2. $|\hat{f}| |\delta| < 1 \ \forall \ \omega$

This is only a sufficient condition for stability. Also note that if $G$ has unstable poles, then $\hat{G}$ should have them too. Otherwise, $\delta$ will be unstable.
Now let’s define a weighting filter for robustness such that

1. \(|W_r| |\delta| \geq 1\), \(W_r\) is stable
2. \(\Delta\) denotes all stable transfer functions such that \(|\Delta| \leq 1 \ \forall \ \omega\)

The robustness model in Figure X now reduces to the ones in Figure 3.

![Figure 3: Reduced Robustness Model](image)

This system is \textit{robustly stable} if and only if

\[ \hat{\|W_r\|} < 1 \]

This is both a sufficient and necessary condition for stability to any arbitrary phase distortion based on the closed-loop transfer function.

We find the \(W_r1\) and \(W_r2\) for the translation and rotation models by upper-bounding the \(\delta_1(\omega)\) and \(\delta_2(\omega)\). The actual plant is treated as the frequency response data obtained from the DSA, and the nominal models are the frequency data from the 3rd order models we curve fitted.

![Figure 4: Experimental Wr](image)
The transfer functions of the $W_r$'s are

$$W_{r_1} = 5 \cdot 10^{-03} \frac{(s + 2000)(s + 800)}{s + 10^4}$$

$$W_{r_2} = 4 \cdot 10^{-04} \frac{(s + 5000)(s + 1000)}{s + 2 \cdot 10^4}$$

We will design the controller $C$ based on $\hat{G}$ such that $1 + \hat{G}C = 0$ is stable, i.e.

$$\hat{T} = \frac{\hat{P}C}{1 + \hat{P}C}$$

is stable. Together with $W_r$, we are able to determine the stability robustness of each controller design.

An important remark of this result is we desire the magnitude of $\hat{T}$ small for more robust stability. However, that would cause $S$ to become larger and affect performance by decreasing the gain and phase margins. Thus, there is a tradeoff between robust stability and performance. At the expense of ensuring the compensated system is stable given modeling errors of the plant, performance is sacrificed.
Selection of Sampling Time

Since we will implement controllers digitally, we will need to consider the significance of our choice of sampling time, $T_s$. For small sampling times, the system becomes susceptible to round-off errors, reducing the accuracy of signals and precision for poles and zeros. And too high a sampling time uses much of the system’s computing resources without significant gain in performance.

To understand the choice of sampling time, the continuous plant is modeled as shown in Figure 1 with zero-order hold block and a sampler.

![Figure 1: Discretization of Plant](image)

The zero-order hold is mathematically represented and approximated as

$$\frac{1-e^{-\pi T_s}}{s} \approx e^{-\frac{T_s}{2}}$$

Depending on the value of $T_s$, this phase correction affects the gain cross over frequency, and therefore changes the system’s phase margin. Typically, we desired this change between 5 and 15 degrees.

$$\frac{180^\circ}{\pi} \cdot \frac{T_s}{2} \cdot \omega_c \cong 5^\circ \sim 15^\circ$$

$$T \omega_c \cong 0.15 \sim 0.5$$

$$\frac{1}{f_s} (f_c \cdot 2\pi) \cong$$

$$\frac{f_s}{f_c} \cong 10 \sim 40$$

Thus, for a 5 degree reduction in the phase margin, we choose a sampling frequency of at least 40 times larger than the gain cross over frequency.

For the magnetic bearing, the gain crossover frequency is at 150 degrees. A sampling frequency of 5000 Hz meets this condition for small phase margin reduction.
CONTINUOUS-TIME (INDIRECT) DESIGN OF LEAD-LAG COMPENSATOR

A lead-lag controller in a negative feedback scheme is considered to stabilize the closed loop step response and minimize its steady-state error. This will be designed using the root locus method using MATLAB’s `rlocus` and `sisotool` commands. We consider the translational model in this controller design, and later find out that this will also stabilize the rotational model.

![Figure 1: Root Locus of (a) Uncompensated System and (b) Lead-Lag Compensated System](image)

The poles and zeros of the lead controllers were first selected at strategic locations along the real axis such that the poles would breakaway and head towards zero while remaining in the left half plane as the gain increased. The pole/zero placements for this controller have implications to the transient characteristics of the system’s step response. The real-time LTI viewer allows for an “on-the-fly” observation as the placements were tweaked.

![Figure 2: Zoomed-in Root Locus of Lead-Lag Compensated](image)
To improve steady-state error of the type 0 system, a lag compensation was employed. Because the uncompensated (and compensated) system is type 0, it will have a steady-state error. The pole was chosen at 0, so it functions like an integrator to eliminate steady-state error, and the zero was chosen close to the pole have a small effect on the DC gain.

Lastly, the compensator gain is adjusted by moving the poles along the root locus. This was performed with consideration to the following characteristics of the step response.

- Peak Amplitude
- Rise Time
- Settling Time

The transfer function of the designed controller is

\[
C(s) = 6.2829 \frac{(1 + 0.0028s)(1 + 0.086s)}{s(1 + 0.0018s)}
\]

The open loop and the closed loop lead-lag compensated step responses are shown in Figure 3.

![Figure 3: Step Responses of (a) Open Loop System and (b) Compensated System](image-url)
Stability margins of the designed compensated system can be examined by plotting the open loop Bode plot, shown in Figure 4.

In determining how close the system is to instability, the relevant gain margin is about 14 dB or K = 5.0119 and the phase margin is about 78 degrees. The phase margin is exceptionally good with the gain margin being less so.
Discretization of Continuous-time Controller

The bilinear transform approximation is used to find the discrete-time equivalent of the lead-lag compensator. Using MATLAB’s `c2d` command with a realizable sampling time of \( T_s = \frac{1}{5000/Hz} = .0002 \text{sec} \), the discretized controller becomes

\[
C(z) = C(s) \bigg|_{s = \frac{2}{T_s}} = \frac{0.82567(z - 0.9977)(z - 0.931)}{(z - 1)(z - 0.8947)}
\]

The frequency response of the continuous and discrete time controllers is shown in Figure 5.

![Figure 5: Frequency Response of Continuous and Discrete Time Controllers](image)

**Figure 5:** Frequency Response of Continuous and Discrete Time Controllers

![Figure 6: Comparison of Open loop Bode of Continuous-time and Discrete-time Compensated System](image)

**Figure 6:** Comparison of Open loop Bode of Continuous-time and Discrete-time Compensated System
The approximation begins to worsen significantly at $800 \text{ rad/sec}$ for the phase plot. This frequency at which distortion becomes more significant may need to be larger depending on the frequency range of interest. Increasing the sampling time would make the adjustment. Note that the stability margins in the continuous-time system are almost identical to the ones in the discrete-time system. Thus, the indirect design of the controller, although an approximation at the discretization step, maintains the system stability characteristics, and should produce similar results in implementation as predicted in continuous-time design.
DISCRETE-TIME (DIRECT) DESIGN OF LEAD-LAG CONTROLLER

The direct controller design involved discretizing the plant via the bilinear transform and using MATLAB’s *sisotool* command to place poles and zeros. Figure 1 shows the root locus of the plant in the $z$-domain.

![Root Locus of G(z)](image)

*Figure 1: Root Locus of G(z)*

Because most of the root locus lies outside the unit circle and results in system instability, we place poles and zeros inside the unit circle to bend the root locus to the left to lie inside the stable region. An initial attempt is to place the poles/zeros directly on top of the ones furthest away from the origin. This would result in a lead compensator since $p > z$. We place another pair of pole and zero in a lag configuration to eliminate steady-state error. The resulting root locus is shown in Figure 2.

![Root Locus of Lead-lag Directly Compensated System](image)

*Figure 2: Root Locus of Lead-lag Directly Compensated System*

A point on the root locus corresponding to a gain $K$ was selected based on how it affected the step response. Consideration in this selection was based on the peak amplitude of the response, rise time, and settling time. The compensator chosen is

$$C_{\text{lead–lag}}(z) = 1.7762 \frac{(z - 0.9202)(z - 0.998)}{(z - 0.7563)(z - 1)}$$
The step response of this closed-loop compensated system is shown in Figure 3.

![Step Response](image)

**Figure 3**: Step Response of Direct-Design Compensated System

Table 1 compares the stability margins in both the direct and indirect design of the lead-lag controllers.

<table>
<thead>
<tr>
<th></th>
<th>Indirect</th>
<th>Direct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain Margin [dB]</td>
<td>6.77</td>
<td>7.51</td>
</tr>
<tr>
<td>Phase Margin [degrees]</td>
<td>9.02</td>
<td>22.8</td>
</tr>
</tbody>
</table>

**Table 1**: Comparison of Stability Margins

Using MATLAB’s *sisotools* in both designs, the direct method is better suited for achieving desired stability margins because of the real-time design capability of modifying the controller and observing the effects of these margins. In indirect design, the stability margins were designed for in continuous-time, and deviates from the designed figures as a result of the bilinear transform approximation. This is especially noticeable in the phase margin.
Note the roll-off in the sensitivity functions that indicates the existence of an integrator in the compensator. This is expected since the lead-lag effectively tracks step references with zero steady-state error.

The complementary sensitivity function tells us the bandwidth of the system, which is roughly at 1000 rad/s in both translation and rotation.
**Robust Stability Analysis**

The indirect Lead-Lag compensated system is not robustly stable since it is the plotted curve is not less than 0dB for all frequencies.

Similarly, the direct Lead-Lag compensated system is not robustly stable.
Simulation

Prior to applying the designed controller to the MBC500, we perform several simulations in MATLAB and Simulink to gauge performance. This gives confidence of our controller’s performance and reduces risk of destabilizing the magnetic bearing or saturating the current and voltage sources. Figure 4 is the Simulink block diagram to run the simulations.

![Simulink Block Diagram](image)

**Figure 4:** Simulink Simulation of Direct and Indirect Lead-Lag Compensated 10th Order Plant

A lead-lag compensator was designed both directly and indirectly using the decoupled third order system representation. Stability and performance of the controllers were analyzed, and resulted in a pair of C(z)’s given in the design sections.

The plant and indirect controller are both discretized using the continuous-time plant and controller version in preparation for digital implementation. In the case of the direct design controller, only the plant needed discretization. A realizable sampling frequency of 5000 Hz was used in generating the zero order hold plant and the bilinear transform of the controller. The root locus and bode plots were examined in the previous section to ensure the discretized plant/controller pair resulted in a stable system.

The 10th order plant replaces the 3rd order plant since it models the dynamics of the system more closely. Although the higher order of the system cannot be explained on a physical basis, the simulation will provide insight to how the system will perform realistically. The Simulink model is used for simulation. Figure 5 show the y-direction system outputs due to step inputs with a half second delay between them. Because the controller was not designed with saturation limits in mind, the gains of the controllers were modified so that the output is within the limits of +5 Volts.
The modified controllers have slightly increased gains, and are given below.

\[ C_{y_1} = 1.12567 \frac{(z - .9977)(z - .931)}{(z - 1)(z - .8947)} \]

\[ C_{y_2} = 0.82567 \frac{(z - .9977)(z - .931)}{(z - 1)(z - .8947)} \]

**Figure 5:** Step Response of Discretized Indirect Lead-lag Compensated 10th Order Plant – y

**Figure 6:** Step Response of Discretized Indirect Lead-lag Compensated 10th Order Plant – u
Figure 7: Step Response of Direct Lead-lag Compensated 10th Order Plant – $y$

Figure 8: Step Response of Direct Lead-lag Compensated 10th Order Plant – $u$

Note that tracking to the step reference given some time for the transients to settle. The controller outputs are also examined to ensure the signal stays within the saturation limits.
Implementation

Once the previous steps convince us that the controller will stabilize the plant with desired performance characteristics in simulation, we implement the controller with the magnetic bearing apparatus. The following figures show the step responses of $y_1$ and $y_2$ with equal magnetic forces applied in the same direction at both shaft ends for translation, and in opposite directions for rotation.

The experimental and simulated step responses (Step 3 and 4) are shown in Figures 13 through 16 for both controllers. Note that two step responses, with the same amplitude and different delays, are applied at both sensors to reveal the degree of decoupling the physical system exhibits.

![Graph](Image1.png)

**Figure 13.** Step Response of System Output with Direct Design Lead-Lag Controller

![Graph](Image2.png)

**Figure 14.** Step Response of Controller Output with Direct Design Lead-Lag Controller
Note that Figures 13 through 16 signify that our lead-lag controllers are correctly implemented and tracks a step response close to zero error as expected from simulations. There are slight differences for the indirectly designed controller—more so than the direct designed one—in the simulated and experimental step responses. Although the peak amplitude and transient ringing are somewhat matching, these differences can be attributed to the approximation of the continuous-time controller used in the indirect design, namely the distortion from frequency warping.

In both controller designs, note that a step input applied at a sensor location has noticeable effects on the other sensor, signifying the system’s translational and rotational dynamics are slightly coupled. Our model assumes perfect decoupling, but the experimental results indicate dynamics with relatively small magnitudes.

One important check at this step is to examine the intersample interpolation. Essentially, we need to sample at a sufficiently high sample rate to avoid aliasing and incorrect conclusions of the discretized output signals. At the sampling frequency of 5 kHz, our simulation and experimental data appear free of aliasing.
Sinusoidal Reference Tracking

Supposed the system receives an input other than the step reference that the lead-lag compensator does so well in tracking. We run simulations with sinusoidal inputs applied to both the translational and rotational directions, and check the tracking performance.

![Simulation Output: Ytranslation](image1)

![Simulation Output: Yrotation](image2)

**Figure 17:** Sinusoidal Reference Tracking with Direct Lead-Lag Controller

**System Output**

![Simulation Control: Utranslation](image3)

![Simulation Control: Urotation](image4)

**Figure 18:** Sinusoidal Reference Tracking with Direct Lead-Lag Controller

**Controller Output**

Although the system output has the same oscillation frequency as the reference, the directly designed lead-lag compensator does not track sinusoidal references with zero steady-state error.

We will consider other controllers that can accomplish sinusoidal and eventually periodic reference tracking.
STATE OBSERVER FEEDBACK CONTROLLER

This controller leverages modern control design techniques, in other words, state-space form. The graphical representation that classical design features is not apparent in modern design. However, designers have a richer mathematical framework for placing closed-loop system poles by designing a feedback gain for achieving that.

Often times, state variables are not available for control design. Estimation techniques are thus needed. The Luenberger Observer is explored and trialed in constructing a state feedback controller for the MBC500.

Formulation

A controller for the system \((A,B)\) can be designed using state feedback if and only if \((A,B)\) is controllable. The locations of the system’s closed-loop poles can be placed anywhere with the appropriate state feedback gain \(K\). We can represent the system in the state-space form

\[
x(k+1) = Ax(k) + Bu(k) \\
y(k) = Cx(k)
\]

with the state feedback and feed-forward control law

\[
u(k) = -Kx(k) + Nr(k)
\]

With the control law, the closed-loop system is

\[
x(k+1) = (A - BK)x(k) + BNu(k) \\
y(k) = Cx(k)
\]

The associated transfer function is

\[
Y(z) = C[zI - (A - BK)]^{-1} BNr(z)
\]

To examine the phase and gain margins of the system, we look at the characteristic equation

\[1 + K(zI - A)^{-1} Bu(z) = 0\]

where the loop gain is

\[L(z) = K(zI - A)^{-1} B\]
The gain $K$ can be chosen to have the specified closed-loop poles using MATLAB’s `place` command with the matrices $(A,B)$, and vector $P$ containing the desired poles.

State feedback relies on the fact that information of the states is available. In practice, this is often not the case. We can augment the state feedback controller with state estimation. The closed-loop Luenberger observer includes extensive information of the states from output measurements. In this scheme, we examine the error dynamics defined as

$$\tilde{x}(k) = x(k) - \hat{x}(k)$$

We desire the error dynamics to converge to 0 in the steady-state so that our estimated states are close to the actual ones. The observer state-space equation is

$$\tilde{x}(k+1) = A\tilde{x}(k) - LC\hat{x}(k) = (A - LC)\tilde{x}(k)$$

The observer gain $L$ is usually chosen to place the closed-loop poles such that the state estimation is quicker than the state feedback. The observer poles can be placed at any location if and only if $(A, C)$ is observable. MATLAB’s `place(A’,C’,P)` command is often used for pole placement. The state estimation feedback can be written as the following:

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & -BK \\ 0 & A-LC \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} N \cdot r(t)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix}$$

The closed-loop characteristic equation of the system above is given by

$$\det \begin{bmatrix} zI - (A - BK) & BK \\ 0 & zI - (A - LC) \end{bmatrix} = \det[zI - (A - BK)] \cdot \det[zI - (A - LC)]$$

This shows that $n$ of the closed-loop eigenvalues, or poles, are from the state feedback design and the other $n$ eigenvalues are from the observer compensator design. This highlights the separation principle i.e. the state feedback control poles can be designed separately from the observer poles.

To examine the phase and gain margins of the modified system, we look at the characteristic equation

$$1 + K [zI - (A-LC-BK)]^{-1} LC(zI - A)^{-1} B = 0$$
where the modified loop gain is

\[ L(z) = K \left[ zI - (A - LC - BK) \right]^{-1} LC(zI - A)^{-1}B \]

**Internal Model Augmentation**

For reference tracking with zero steady-state error, we can include a form of the reference signal into the controller. This is called Internal Model Control. For step reference tracking, we can build integral control into this controller through a state augmentation. We accomplish this by augmenting the model of the plant with an integrator, which adds an error integral output to the existing plant output. We define the error signal as

\[ e(k) = y(k) - \hat{y}(k) = C(x(k) - \hat{x}(k)) \]

The propagated integral of the error is formulated as

\[ e(k+1) = e(k) + [y(k) - r(k)] = e(k) + [Cx(k) - r(k)] \]

The augmented state equation is then

\[
\begin{bmatrix}
    x(k+1) \\
    e(k+1)
\end{bmatrix} =
\begin{bmatrix}
    A & 0 \\
    C & 1
\end{bmatrix}
\begin{bmatrix}
    x(k) \\
    e(k)
\end{bmatrix} +
\begin{bmatrix}
    B \\ 0
\end{bmatrix} u(k) +
\begin{bmatrix}
    0 \\ -1
\end{bmatrix} r(k)
\]

\[ = A' \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} + B'u(k) + N'r(k) \]

The modified control gain has the following form

\[ K = [K_i | K_f] \]

where \( K_i \) is the added element is the gain for the integrator. As before, the place command with the new \( A' \) and \( B' \) matrices can be used to determine the feedback gain. The block diagram of this state-space system is shown in Figure 1.
The equation for the internal model is

\[ \dot{x}_D = A_D x_d - B_D C + r \]

For the integrator,

\[ \dot{x}_D = x_d + r - K_i C \]

We augment the state-space system once more to get

\[
\begin{bmatrix}
  x(k+1) \\
  x_d(k+1) \\
  e(k+1)
\end{bmatrix} =
\begin{bmatrix}
  A & 0 & 0 \\
  -B_D C & A_D & 0 \\
  0 & 0 & A - LC
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  x_d(k) \\
  e(k)
\end{bmatrix} +
\begin{bmatrix}
  B \\
  0 \\
  0
\end{bmatrix}
\begin{bmatrix}
  u(k) \\
  -1 \\
  0
\end{bmatrix}
+ r(k)
\]

\[
= \begin{bmatrix}
\bar{A} \\
0 \\
A - LC
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  x_d(k) \\
  e(k)
\end{bmatrix} +
\begin{bmatrix}
\bar{B} \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
  u(k) \\
  -1 \\
  0
\end{bmatrix}
+ r(k)
\]

\[
= A^* x(k) + B^* u(k) + N^* r(k)
\]

The modified control law has the following form

\[
u = [-K \ K_D]
\begin{bmatrix}
  x(k) \\
  x_d(k)
\end{bmatrix} =
\begin{bmatrix}
  x(k) \\
  x_d(k) \\
  e(k)
\end{bmatrix} =
\begin{bmatrix}
\bar{K} \ K
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  x_d(k) \\
  e(k)
\end{bmatrix}
\]
The closed loop “A” matrix is \( A - BK \):

\[
A'' - B'K = \begin{bmatrix} \bar{A} & 0 \\ 0 & A - LC \end{bmatrix} - \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} \begin{bmatrix} K \\ \bar{K} \end{bmatrix} = \begin{bmatrix} \bar{A} - \bar{B} \bar{K} & \bar{B} \bar{K} \\ 0 & A - LC \end{bmatrix}
\]

We arrive at an upper triangular A-matrix once again, so the separation principle also applies in determining feedback and observer gains.

For robust stability analysis, we need to calculate the sensitivity and complementary sensitivity functions.

The loop gain from the integrator output \( v \) to \( y \) is

\[
L_i(z) = \frac{1}{1 + K(zI - A + LC)^{-1}B} \cdot G(z) \cdot \left( K(zI - A)^{-1}L + \frac{K_i}{z-1} \right)
\]

The derivation is provided in the Appendix.

The sensitivity and complementary sensitivity functions for the integrator system are

\[
S = \frac{1}{1 + L(z)} = \frac{1 + K(zI - A + LC)^{-1}B}{1 + K(zI - A + LC)^{-1}B + G(z) \cdot \left( K(zI - A + LC)^{-1}L + \frac{K_i}{z-1} \right)}
\]

\[
T = 1 - S = \frac{G(z) \cdot \left( K(zI - A + LC)^{-1}L + \frac{K_i}{z-1} \right)}{1 + K(zI - A + LC)^{-1}B + G(z) \cdot \left( K(zI - A + LC)^{-1}L + \frac{K_i}{z-1} \right)}
\]

The \( T \) is also used in robust stability analysis.

The characteristic equation of this system is now

\[
1 + L_i(z) = 0
\]

\[
1 + \frac{1}{1 + K(zI - A + LC)^{-1}B} \cdot G(z) \cdot \left( K(zI - A + LC)^{-1}L + \frac{K_i}{z-1} \right) = 0
\]
Design

The state observer feedback controlled system will have the block diagram shown in Figure 2.

We begin with the third order discretized transfer functions for translation and rotation, and obtain their associated \(A, B, C,\) and \(D\) matrices. Because the two systems have similar input-output relationships, we design the same controller for each. By the separation principle, we design for the \(K\) and \(L\) matrices independently. We arbitrarily place the closed-loop feedback poles at

\[
\text{poles} = [0.88, 0.83, 0.8];
\]

for stability and quick transient dynamics. Using these pole locations with the \textit{place} command in MATLAB, the following \(K\) matrices were obtained:

\[
K_t = \begin{bmatrix}
-0.1285 & 0.2136 & -0.1417 \\
\end{bmatrix}
\]

\[
K_r = \begin{bmatrix}
-0.1132 & 0.2049 & -0.1400 \\
\end{bmatrix}
\]
We arbitrarily place the observer poles at

\[ \text{poles}_\text{obs} = [.68, .63, .6] \]

which produces the corresponding gain matrices

\[
\begin{bmatrix}
36.7192 & 62.3149 & 51.7825 \\
21.1890 & 35.3741 & 28.9363
\end{bmatrix}
\]

The forward gain matrices \( N_t \) and \( N_r \) are calculated so that the input and output are scaled the same. This is achieved by solving the equation

\[
N = \left( C \left[ I - A + BK \right]^{-1} B \right)^{-1}
\]

The integrator gain is arbitrarily chosen at 25 for the augmented controller that resulted in good transient performance.
Sensitivity and Complementary Sensitivity Analysis

Note the increased performance from the lead-lag compensator. The bandwidth of the sensitivity transfer function is much higher.

**Figure 3:** State-Feedback Controller

With the augmented integrator, note the roll-off at low frequencies that we expect.

**Figure 4:** State-Feedback with Integrator Controller
Robust Stability Analysis

Note that the state-feedback controller, as designed, is not robustly stable in either direction.

Figure 5: State-Feedback Controller

Figure 6: State-Feedback with Integrator Controller
Simulation

Using MATLAB and Simulink, the state observer feedback controllers for the third order models for translation and rotation were created. Note the existence of non-zero steady-state error to pulse step reference.

The controller output signals are also shown in Figure 8 to verify that they are not saturating the plant.
Figure 9 show the output of the system compensated with the integral state estimation controller. Note the elimination of steady-state error in the reference tracking of step inputs.

![Simulation Output: Y_transformation](image1)

![Simulation Output: Y_rotation](image2)

**Figure 9:** Simulated Pulse Response of Augmented State Estimator Feedback Control, Output

Again, the controller output signals are shown, and do not saturate the plant.

![Simulation Control: U_transformation](image3)

![Simulation Control: U_rotation](image4)

**Figure 10:** Simulated Pulse Response of Augmented State Estimator Feedback Control, Controller
Implementation

Figures 11 and 12 show the system and controller output signals without integral control. Note the existence of steady-state error to step reference tracking and the coupling effects between translation and rotation.

![Figure 11: Implemented Step Response of State Estimator Feedback Control, Output](image1)

![Figure 12: Implemented Step Response of State Estimator Feedback Control, Controller](image2)
Figure 13 and 14 shows the system and controller output signals with integral control. Note the elimination of steady-state error to step reference tracking.

**Figure 13**: Implemented Step Response of Augmented State Estimator Feedback Control, Output

**Figure 14**: Implemented Step Response of Augmented State Estimator Feedback Control, Controller

**Summary**

As shown, state estimator feedback provides improved bandwidth and tracking performance than that achieved by lead-lag compensator. The theory of this modern control technique affords a richer mathematical framework for achieving closed-loop poles, an improvement over the more free-form, and graphical procedure in classical design.

In the design result, the tracking of a step reference delivers less transient response and quicker regulation. The augmentation feature permits application of IMP. Hence, tracking of sinusoidal inputs is also possible with ease.
POLE PLACEMENT AND MODEL MATCHING CONTROLLER (RST)

The goal of this design is to present a control technique using manipulations of polynomials to match a user-defined model. We specify polynomial transfer functions $R$, $S$ and $T$ that result in a desired closed-loop input-output relationship. Again, we are interested in step and sinusoidal reference tracking. We compare its performance to that of the lead-lag and state estimator feedback controllers.

The design formulation assumes the variables are polynomials and applicable to both continuous-time and discrete-time frameworks. We also impose two assumptions for the plant:

1. Polynomials $A$ and $B$ are coprime.
2. $A$ is monic.

![Figure 1: Block Diagram for RST Controller](image)

**Formulation**

This controller approach has two degrees of freedom, namely feedback and feed-forward. The input-output transfer function for the diagram in Figure 1 is

$$\frac{Y}{U} = \frac{BT}{AR + BS} = \frac{B_m}{A_m}$$

Using the appropriate $R$, $S$, and $T$, we wish the transfer function to have the form of $\frac{B_m}{A_m}$.

For the purpose of causality in digital control, it is required that

$$\deg[R(q)] \geq \deg[S(q)]$$
$$\deg[R(q)] \geq \deg[T(q)]$$

We let $R = \tilde{R}D$ where $D$ may be any polynomial of our choosing such as an integrator as will select later to remove steady-state error to step references.

We first consider the zeros of the closed-loop transfer function. We note that we cannot move the plant zeros like we do plant poles. The zeros can only be cancelled. Unstable ones will always remain in the closed-loop dynamics; hence adding poles to cancel these.
will result in them being unstable. Thus, we separate the unstable and stable zeros by defining

\[ B(q) = B^+(q)B^-(q) \]

where \( B^+ \) contains the stable zeros and is monic, and \( B^- \) contains the unstable zeros of \( B \). The transfer function now has the form

\[
\frac{Y}{U} = \frac{B^+ B^T}{A\bar{R}DB^+ + B^+ B^- S} = \frac{B^+ (B^T)}{B^+ (A\bar{R}D + B^- S)} = \frac{B^- \bar{B}_m}{A_m} \cdot \frac{A_o B^+}{A_o B^+}
\]

where \( A_o \) represents the cancelled observer dynamics. We note that our polynomials we seek are

\[
R = \bar{R}DB^+ \\
T = \bar{B}_m A_o
\]

Comparing denominator polynomials, we get the Diophantine Equation

\[
A\bar{R}D + B^- S = A_mA_o
\]

We can solve this equation using the Sylvester Matrix for \( \bar{R} \) and \( S \).

The closed-loop poles are

\[
A_mA_o B^+
\]

The designer has freedom to define \( A_m \) and place the closed-loop poles wherever desired.

In this design, we must ensure the following conditions are true for system causality:

\[
\deg[S] = \deg[A] + \deg[D] - 1 \\
\deg[A_o] = 2\deg[A] - \deg[A_m] - \deg[B^+] + \deg[D] - 1 \\
\deg[A_m] - \deg[B_m] \geq \deg[A] - \deg[B]
\]
For stability margin analysis, we define the sensitivity and complementary sensitivity functions

\[ T_{reg} = \frac{BT}{AR + BS} \]
\[ S_{reg} = \frac{1 - BT}{AR + BS} \]

For robustness analysis, we define the robust sensitivity function

\[ T_r = \frac{BS}{AR + BS} \]

**Design**

We start with the third order decoupled transfer functions for translation and rotation.

\[ G_{d11} = \frac{-0.08836(z - 1.399)(z - 0.7597)}{(z - 1.081)(z - 0.9199)(z - 0.4453)} \]
\[ G_{d22} = \frac{-0.69332(z - 1.707)(z - 0.7255)}{(z - 1.092)(z - 0.918)(z - 0.4431)} \]

Using the design guidelines, the relevant polynomials for these models are

\[ B_{ut} = z - 1.399 \quad B_{wr} = z - 1.707 \]
\[ B_{c11} = -0.88641z - 0.06734 \quad B_{cr} = -0.06933z - 0.0503 \]
\[ A_{ut} = 1 \quad A_{ur} = 1 \]
\[ A_{ct} = (z - 1.081)(z - 0.9199)(z - 0.4453) \quad A_{cr} = (z - 1.092)(z - 0.918)(z - 0.4431) \]
\[ = z^3 - 2.446z^2 + 1.885z - 0.4428 \quad = z^3 - 2.453z^2 + 1.893z - 0.4442 \]
\[ \overline{B}_m = -.0075 \quad \overline{B}_m = -0.0042 \]
\[ A_n = z^2 - 1.89z + 0.893 \quad A_m = z^2 - 1.89z + 0.893 \]
\[ A_p = z^3 - 1.95z^2 + 1.265z - 0.273 \quad A_o = z^3 - 1.95z^2 + 1.265z - 0.273 \]
\[ D = z + 0.3 \quad D = z + 0.3 \]
\[ E = 1 \quad E = 1 \]
The corresponding RST transfer functions for these design choices are

\[
R_i = \frac{-z^3}{0.08864z^3 - 0.148z^2 + 0.05333z + 0.00605} \\
S_i = \frac{-0.4838z^3 + 1.145z^2 - 0.8644z + 0.2027}{z^3} \\
T_i = \frac{-0.007519z^3 + 0.01466z^2 - 0.009511z + 0.002053}{z^3}
\]

and

\[
R_r = \frac{-z^3}{0.06933z^3 - 0.1286z^2 + 0.06579z + 0.006514} \\
S_r = \frac{-0.2575z^3 + 0.6108z^2 - 0.4626z + 0.1091}{z^3} \\
T_r = \frac{-0.004243z^3 + 0.008274z^2 - 0.005368z + 0.001158}{z^3}
\]

We also wish to have the system track sinusoidal references. We accomplish this by choosing \( D \) to have the form

\[
D = z^2 - 2 \cos(\omega_b T_s) z + 1
\]

so that it is similar to the \( z \)-transform of the input. This is the Internal Model Principle. Doing this, we also need to increase the degree of the observer dynamics by one to satisfy the controller condition.

We need to modify our block diagram so that we can define an error signal to eliminate.

![Figure 2: Modified RST Controller](image)

We only need to use the pole placement technique to achieve sinusoidal reference tracking, i.e. obtain \( S \) and \( R \). We make \( T = 1 \) for simplicity.
Sensitivity and Complementary Sensitivity Analysis

Note the RST controller without integrator \((D = 1)\) shows the magnitude sensitivity much higher than 0 dB for low frequencies. Thus, we can expect this design to track step references with steady-state error. In our case, there will be significant error in the translation and slightly less for rotation.

Adding an integrator produces a roll-off in the magnitude of the sensitivity transfer function for low frequencies. Hence, we can expect tracking to step references.

Figure 3: RST Controller, \(D = 1\)

Figure 4: RST Controller, \(D = z - 1\)
In tracking a sinusoidal reference, we wish to have a notch at the reference’s frequency. This ensures mathematical cancellation of the signal dynamics. Figure 5 indicates a notch at a 100 rad/s. We should expect tracking of sinusoidal references with this frequency content.

**Figure 5:** Modified RST Controller, $D = z^2 - 2 \cos(\omega_0 T_z) z + 1$
Robust Stability Analysis

The RST compensated with system with $D = 1$ is not robustly stable in translation, but is in rotation.

The RST compensated with system with $D = z - 1$ is robustly stable.
The RST compensated with system with \( D = z^2 - 2 \cos(\omega_0 T_v) z + 1 \) is not robustly stable.
Simulation

Figures 9 and 10 show the simulated output and control signals with pulse inputs. Note the existence of steady-state error in reference tracking of the input.

**Figure 9:** Simulated Pulse Response of RST Controller, Output

**Figure 10:** Simulated Pulse Response of RST Controller, Controller
Figures 11 and 12 show the simulated output of RST compensation with an integrator. The choice of $D = z^{-1}$ achieves this, and results in zero steady-state error.

Figure 11: Simulated Pulse Response of RST Integrator Controller, Output

Figure 12: Simulated Pulse Response of RST Integrator Controller, Controller
And the simulation for sinusoidal reference tracking of a 15 Hertz signal. Note this is approximately 100 rad/s where the notch frequency appeared in the sensitivity Bode magnitude plot.

**Figure 13:** Simulated Sinusoidal Response of RST Controller, Output

**Figure 14:** Simulated Sinusoidal Response of RST Controller, Controller
Implementation

Figures 15 and 16 show the experimental output and control signals with step inputs.

Note the existence of steady-state error to reference tracking of the step input, and the coupling effects.
To remove the steady-state error to step references, an integrator was added. Figures 17 and 18 shows that we accomplish this by choosing $D$ to have the integrator form

$$D = z - 1$$

**Figure 17:** Implemented Step Response of RST Integrator Controller, Output

**Figure 18:** Implemented Step Response of RST Integrator Controller, Controller
To remove the steady-state error to sinusoidal references, an integrator was added. Figures 19 and 20 shows that we accomplish this by choosing D to have the integrator form

\[
D = z^2 - 2 \cos(w_0 T_s) z + 1
\]

**Figure 19:** Implemented Step Response of RST Sinusoidal Internal Model Controller, Output

**Figure 20:** Implemented Step Response of RST Sinusoidal Internal Model Controller, Controller

**Summary**

The polynomial RST design approach is another mathematically-rich framework for controller design. It allows the user to specify a final transfer function for a compensated system. Via pole-placement and model matching through solving the Diophantine Equation, the final system is realized.

Internal modeling is also permitted, affording reference tracking as long as the dynamics of the input are known and inserted in the feedback path.
**$H_2$ AND $H_{\infty}$-NORM CONTROLLER**

We wish to design a controller that can reject disturbances in its output signal. In other words, we want to minimize the transfer function from $d$ to $y$ in Figure 1.

A compensated system’s sensitivity function describes its ability to reject disturbances over a range of frequency. This sensitivity function for the block diagram in Figure 1 is

$$S = \frac{1}{1 + KG}$$

The controller we seek minimizes this sensitivity. The best controller, as we will discover, has minimum variance. In control applications, this is equivalent to minimizing the 2-norm of the system. As shown in the Robust Stability Analysis Framework section, the peak of the sensitivity function is the infinity-norm of that function. We also seek to minimize this as well for good stability margins.

**Formulation**

In minimizing $S$, we consider minimizing the 2-norm and infinity-norm. Graphically, the 2-norm can be interpreted as the area under the sensitivity function while the infinity-norm is its maximum peak. The weighting function $W_p$ can specify the frequency bands of most interest in the minimization. Thus, the minimization function is

$$\|SW_p\|_{(\infty)}$$

The design of the stabilizing controller $K$ is made easier by adopting the framework of Youla Parameterization of stabilizing controllers. The block diagram is shown in Figure 2.
Applying the Small Gain Theorem, as long as $Q \in RH_\infty$ (the class of stable controllers) and $\hat{G}$ is stable, then the closed-loop is also stable. If the system is unstable, we can choose a simple controller $C$ to stabilize the original plant $G$ in a feedback fashion and use this closed-loop compensated system as $\hat{G}$. The sensitivity function from $d$ to $y$ can be shown to be

$$S = W_p \frac{1 - \hat{G}Q}{1 + Q(G - \hat{G})}$$

Thus we want to minimize

$$\left\| SW_p \right\| = \left\| W_p \frac{1 - \hat{G}Q}{1 + Q(G - \hat{G})} \right\|$$

With the approximation $\hat{G} \approx G$, we have

$$\left\| SW_p \right\| = \left\| W_p - W_p \hat{G}Q \right\| = \left\| \hat{t}_1 - \hat{t}_2 Q \right\|$$

Next, we solve the model matching problem with $t_1 = W_p$ and $t_2 = W_p \hat{G}$. The block diagram
Design

Because our third order models of the translational and rotational plant $G$ are unstable, we stabilize it with a simple controller e.g. a lead compensator. We use the feedback closed-loop transfer function as the stable plant model. The chosen lead compensators are

\[
C_t = 3.5805 \frac{z - 0.8394}{z - 0.4817} \\
C_r = 3.0817 \frac{z - 0.835}{z - 0.485}
\]

The performance weighting filter is chosen as a low-pass filter since we are more interested in the lower frequency band. Application to the magnetic bearing is low frequency based. Also, the higher frequency bands are less understood as evidenced in our analytical modeling.

The defining parameters for the weighting filter are the cross-over frequency, low-frequency gain, high-frequency gain.

We are now able to define our $t_1$ and $t_2$ for norm minimizing. The $h2syn$ and $hinfsyn$ toolkits in MATLAB are used.

The $H_2$ minimized norms are

\[
\gamma_t^{H_2} = 0.4094 \\
\gamma_r^{H_2} = 0.3975
\]

The minimized $H_{\infty}$ norms are

\[
\gamma_t^{H_{\infty}} = 0.6073 \\
\gamma_r^{H_{\infty}} = 0.5200
\]
Sensitivity and Complementary Sensitivity Analysis

Figure 3: $H_2$ Norm Controller

Note that the sensitivity of the $H_2$ Norm Controller is minimized across all frequencies. Because it does not roll-off for lower frequencies, we can expect steady-state tracking error to step references.

And in the case of the $H_\infty$ Norm Controller, the peak of the sensitivity is smaller than that for the $H_2$ controller and that for the previously designed controllers.

The bandwidth of these two controllers can be modified by adjusting the parameters of the weighting filter.

Note that two methods were used to compute the sensitivity transfer function, and they match well.
**Robust Stability Analysis**

The $H_2$ norm compensated system is robustly stable.

The $H_\infty$ norm compensated system is not robustly stable.
Simulation

Figure 7: Simulated Step Response of $H_\infty$ Norm Controller, Translational Output
Entire Trial (left), Zoomed-In (right)

Figure 8: Simulated Step Response of $H_2$ Norm Controller, Rotational Output
Entire Trial (left), Zoomed-In (right)

Figure 9: Simulated Step Response of $H_2$ Norm Controller, Controller Output

The simulated step response of the $H_\infty$ compensated system was unstable, and its output and controller responses are not shown.
Implementation

Figure 10: Implemented Step Response of $H_2$ Norm Controller, Translational Output
Entire Trial (left), Zoomed-in (right)

Figure 11: Implemented Step Response of $H_2$ Norm Controller, Rotational Output
Entire Trial (left), Zoomed-in (right)

Figure 12: Implemented Step Response of $H_2$ Norm Controller, Controller Output
Entire Trial (left), Zoomed-in (right)
Summary

From the Youla Parameterization of the class of all stabilizing controllers, the $H_2$ and $H_\infty$ controllers were selected based on a selected criteria to minimize.

The $H_2$ norm controller has its advantage in that it minimizes the ratio of the output power to disturbance power. Thus, we expect this system to perform the best of all controllers in the stabilizing class in the presence of output disturbance.

The $H_\infty$ norm controller minimizes the peak of the sensitivity function, and produces a system with the best stability margins.

Note that reference tracking is not possible in these two controllers since the minimization of the sensitivity function is over all frequencies.
ZERO-PHASE ERROR TRACKING FEED-FORWARD CONTROLLER

Up to this point, we have focused on feedback controllers. Their major function is to regulate the system of interest against disturbance inputs, although we have also used them for the output to track input references.

Suppose we have a stable plant. We wish to design a feed-forward block that achieves perfect tracking (zero-phase and zero error) to reference inputs.

![Block Diagram for Feed-forward Controller](image)

*Figure 1: Block Diagram for Feed-forward Controller*

**Formulation**

Suppose we have a stable plant $G = \frac{B(z)}{A(z)}$. Choosing $F = \frac{A(z^{-1})}{B(z^{-1})}$ will generally not be a valid controller since $B$ may include unstable zeros. We modify our choice of $F$ to have the form

$$ F = z^d \frac{A(z^{-1})}{B'(z^{-1})B'(z^{-1})} $$

The issue with causality is now no longer a problem and we have separated the zeros into their cancellable and uncancellable components. We have two options in designing $F$ through pole/zero cancellation techniques.

1. Achieve $F$ with magnitude 1 and unknown phase
2. Achieve $F$ with zero-phase with unknown magnitude

The transfer function using the first option is

$$ FG = z^d \frac{A(z^{-1})}{B'(z^{-1})B'(z^{-1})} \cdot z^{-d} \frac{B'(z^{-1})B'(z^{-1})}{A(z^{-1})} = \frac{B'(z^{-1})}{B'(z)} $$

Note that this scheme only works if $B^{-1}(z)$ does not contain unstable zeros.
The transfer function using the second option relies on the fact that the product of a number and its complex conjugate produces a real result, i.e. zero-phase.

\[ FG = z^d \frac{A(z^{-1})}{B'(z^{-1})} \gamma \left( z^{-1} \right) \gamma \left( \frac{B'(z^{-1})B'(z^{-1})}{A(z^{-1})} \right) = \frac{B'(z^{-1})B'(z^{-1})}{\gamma} \]

By choosing gamma appropriately, we can achieve DC gain of 1, and approximately so for lower frequencies.

\[ \gamma = \left( B^{-1}(1) \right)^2 \]

Note the tracking performance at low frequencies is typically better compared to higher frequencies. At other frequencies, the magnitude can be different than the DC gain. However, we can choose gamma in such a way that the gain at a particular frequency is 1, and achieve tracking for references containing that frequency.
Design

Because our magnetic bearing plant is unstable, we use a simple lead compensator to stabilize it and use it in the feed-forward controller design.

We follow the above steps, and obtain feed-forward controllers $F_{y1}$ and $F_{y2}$ for translation and rotation, respectively. The system Bode plots for the two systems are given below.

![Bode Diagram](image1.png)

**Figure 2**: Bode Plots of Zero-phase Feed-forward Compensated System

Note that the lower frequencies have a magnitude at approximately 0 dB. Tracking will be better here than at higher frequencies where the magnitude is greater than 1. Also note that the phase is linear since we designed our realizable $F$ by introducing delays. And these, of course, are linear phase.

Sensitivity/Complementary Sensitivity and Robust Stability Analysis

Stability of the plant is not affected by feed-forward controllers. One can think of the feed-forward block as a gain on the reference. Hence, the sensitivity and complementary sensitivity functions are not affected. Robustness, which pertains to stability, is also not relevant to this controller design.
Simulation

A low frequency sinusoidal reference was given to the system. According to the Bode magnitude plot, the gain is close to 0 dB, and the phase is zero. Thus, we expect tracking without error, as seen in Figure 3.

**Figure 3:** Simulated Zero-phase Feed-forward Compensated System, System Output
Translation @ 10 Hz, Rotation @ 20 Hz

**Figure 4:** Simulated Zero-phase Feed-forward Compensated System, Controller
Translation @ 10 Hz, Rotation @ 20 Hz
At high frequencies, the gain is no longer unity, and we can expect to see the output as the scaled reference. Note that zero-phase still applies.

Figure 5: Simulated Zero-phase Feed-forward Compensated System, System Output Translation @ 100 Hz, Rotation @ 150 Hz

Figure 6: Simulated Zero-phase Feed-forward Compensated System, Controller Output Translation @ 100 Hz, Rotation @ 150 Hz

The tracking error for the low and high frequency reference inputs are given in Figure 7.

Figure 7: Simulated Zero-phase Feed-forward Compensated System, Tracking Error Low Frequency (left), High Frequency (right)
Implementation

We give apply a low frequency sinusoidal input to the system.

Figure 8: Implemented Zero-phase Feed-forward Compensated System, System Output
Translation @ 10 Hz, Rotation @ 20 Hz

Figure 9: Implemented Zero-phase Feed-forward Compensated System, Controller Output
Translation @ 10 Hz, Rotation @ 20 Hz
And here are the results for a higher frequency sinusoidal input.

Figure 10: Implemented Zero-phase Feed-forward Compensated System, System Output
Translation @ 100 Hz, Rotation @ 150 Hz

Figure 11: Implemented Zero-phase Feed-forward Compensated System, Controller Output
Translation @ 100 Hz, Rotation @ 150 Hz

Figure 12: Implemented Zero-phase Feed-forward Compensated System, Tracking Error
Low Frequency (left), High Frequency (right)
Summary

We achieve better tracking of a reference through the design of a zero-phase controller that inverts the dynamics of the plant. It is important we know the plant well to ensure this tracking, i.e. the pole and zero locations, as they were relied on in the controller formulation.

The advantage of this design is that it is relatively simple compared to the other more mathematically rigorous controllers. The feed-forward feature of this design eliminates the need for feedback that has effects on stability and robustness.

As long as the plant is stable, we can achieve near perfect tracking with the simple zero-phase feed-forward controller.


**REPETITIVE CONTROL**

In RST design, we showed that the compensated system’s ability to track a sinusoidal references well. In zero-phase error tracking, we also demonstrated that ability, via a simple feed-forward controller. In repetitive control, we design based on the zero-phase controller to eliminate both steady-state error and also higher repeating errors from *any* periodic reference signal.

*Formulation*

We begin with the zero-phase compensator that results in system dynamics represented as

\[ FG = \frac{B^*(z^{-1})B^*(z)}{\gamma} \]

Consider the block diagram in Figure 1 closes the loop of our system with a compensator that includes the dynamics of an arbitrary periodic signal.

![Figure 1: Block Diagram for Repetitive Control](image)

\( q \) is a low pass filter that we define later and \( Np \) is the period of the reference signal (also the preview look-ahead step). Note that the controller \( C \) is intended to stabilize the plant \( G \) if it is unstable. We refer to this feedback with \( C \) and \( G \) as \( P \) throughout the formulation.

We wish to design \( F \) such that the feedback system is stable. We will refer to our proposed design as the prototype Repetitive control. By block diagram manipulation, we obtain

![Figure 2: Equivalent Block Diagram for Repetitive Control](image)
By the Small Gain Theorem, the feedback system is stable if

1. \( F \) and \( P \) are stable
2. \( |FP - I| < 1 \iff |I - FP| < 1 \quad \forall \omega \)

since \( z^{-Np} \) is always stable and \( |z^{-Np}| = 1 \). Note that this condition becomes necessary and sufficient i.e. if and only if, when \( Np \) is sufficiently large. When this occurs, \( z^{-Np} \) generates arbitrary phase distortion, and the results in robust stability apply.

From condition 2, we have

\[
1 - \frac{B^{-1}(z^{-1})B^-(z)}{\gamma} < 1
\]

Choosing \( \gamma \) such that

\[
0 < \frac{B^{-1}(z^{-1})B^-(z)}{\gamma} < 2
\]

will satisfy the condition and ensure closed-loop stability.

Recall that the system is robustly stable if and only if

1. \( 1 + PC = 0 \)
2. \( |TW_r| < 1 \iff |T| < \left| \frac{1}{W_r} \right| \quad \forall \omega \)

Let's examine robust stability for repetitive control. The loop gain is

\[
PC = \frac{z^{-Np}}{1 - z^{-Np}} M(z, z^{-1}) \quad \text{where} \quad M = \frac{B^{-1}(z^{-1})B^-(z)}{\gamma}
\]

The complementary sensitivity function is

\[
T = \frac{PC}{1 + PC} = \frac{Mz^{-Np}}{1 - z^{-Np} + Mz^{-Np}}
\]

Because the complementary sensitivity function has unity gain as a function of frequency (recall we are achieving asymptotic regulation where the sensitivity function is zero), condition 2 under the robust stability test is unlikely satisfied for

\[
\left| W_r(\omega) \right| \geq 1
\]
Typical uncertainty bounds, as in the case of our magnetic bearing, have high gain for higher frequencies. Hence, it can be concluded that the proposed repetitive controller is not robustly stable from the high-gain feedback at high frequencies.

Seeking a robustly stable controller, we make a modification by introducing a zero-phase low-pass filter \( Q \). The rationale for this choice is our frequency range of interest is on the lower side and our desire for no phase distortion to performance i.e. the sensitivity function. The controller is now

\[
C\left(z^{-1}\right) = \frac{Q(z, z^{-1})z^{-Np}}{1 - Q(z, z^{-1})z^{-Np}} F\left(z^{-1}\right)
\]

\( Q \) is restricted to have less than unity magnitude. The performance of this controller is given by the sensitivity function

\[
S = \frac{1}{1 + PC} = \frac{1 - Qz^{-Np}}{1 - Qz^{-Np} + MQz^{-Np}}
\]

and the robust stability measure is

\[
T = \frac{PC}{1 + PC} = \frac{MQz^{-Np}}{1 + (M - 1)Qz^{-Np}}
\]

With this \( T \), we wish to design the filter \( Q \) such that condition 2 of the robust stability criteria is satisfied.

It can be shown that if the following three conditions hold, then this condition is satisfied and the system is robustly stable.

\[
0 < M \leq 1 \\
0 < Q \leq 1 \\
|Q| < \frac{1}{W_r}
\]

For \( M \) closer to 1, the system achieves higher performance in terms of faster tracking convergence. However, robust stability is weakened as the magnitude of \( T \) increases. For \( M \) closer to 0, performance decreases while the system is more robustly stable since the magnitude of \( T \) is smaller than one for a higher range of frequencies. This highlights the trade-off between performance and robust stability.

Note that for \( M \) between 0 and 2, the resulting system is stable, but not robustly stable.
**Design**

The block diagram shown in Figure 3 is used for real-time implementation.

![Block Diagram](image)

**Figure 3:** Block Diagram for Real-time Repetitive Control Implementation

The blocks are defined as

\[ A = z^{-N_1} \]
\[ B = z^{-N_p + N_1} \]
\[ C = Q \cdot z^{-N_2} \]
\[ D = F \cdot z^{-N_1} \]

where

\[ N_1 = d + \text{deg} \left[ B^1 (z) \right] \]
\[ N_2 = \text{deg} \left[ Q(z) \right] \]

\( d \) denotes the plant’s inherent non-causality step, \( N_p \) is the external signals period, and \( N_1 \) and \( N_2 \) are delays to ensure the filter and feed-forward blocks are causal.

Our choice of \( Q \) is

\[ Q(z, z^{-1}) = \left( \frac{z + 2 + z^{-1}}{4} \right)^m \]

This is a FIR filter that is allowed to be non-causal provided that appropriately sized delays are included in the controller design. We define this as

\[ Q'(q^{-1}) = Q \cdot q^{-N_2} \]

The design continues in the next section in regards to robust stability.
**Robust Stability Analysis**

Depending on the chosen design parameters, the system can be unstable. As shown in the formulation, robust stability hinges upon the relative magnitudes of the complementary sensitivity transfer function and the inverse of the plant uncertainty bound. Three sets of design parameters were chosen, one resulting in an unstable system one leading to a stable, but not robustly stable system, and one that is robustly stable. Their complementary sensitivity transfer functions are shown in relation to the uncertainty bound in Figure 4.

![Bode Diagram](image)

**Figure 4:** Bode Plot for Repetitive Control Robust Stability Analysis

The green curve is theoretically robustly stable while the red curve is not. However, it is stable as we will see in the simulation and implementation.

The design parameters for the three cases are summarized in the following table.

<table>
<thead>
<tr>
<th></th>
<th>$N_1$</th>
<th>$N_p$</th>
<th>Gamma</th>
<th>$Q'$</th>
<th>$N_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstable</td>
<td>2</td>
<td>500</td>
<td>40</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Stable</td>
<td>2</td>
<td>500</td>
<td>40</td>
<td>$\frac{z^2 + 2z + 1}{z^2}$</td>
<td></td>
</tr>
<tr>
<td>Robustly Stable</td>
<td>2</td>
<td>500</td>
<td>40</td>
<td>$\left(\frac{z^2 + 2z + 1}{z^2}\right)^{40}$</td>
<td>40</td>
</tr>
</tbody>
</table>

**Table 1:** Repetitive Control Designs
Simulation

Unstable Q

The following design parameters produce an unstable system.

No results are shown since the simulation file reports a singularity in the solution.

Non-Robust Stable Q

The following parameters result in a stable, but not robustly stable system.

Figure 5: Repetitive Control Simulated Reference Tracking for Translation System (left) and Controller (right) Output

Figure 6: Repetitive Control Simulated Tracking Error Entire Trial (left) and Zoomed-in (right)
Robustly Stable $Q$

Figure 7: Repetitive Control Simulated Reference Tracking for Translation System (left) and Controller (right) Output

Figure 8: Repetitive Control Simulated Tracking Error Entire Trial (left) and Zoomed-in (right)
Implementation

We give the translational input a triangular wave, which contains several frequencies, and observe the tracking performance. Input in the rotational direction is not given.

We observe the controller output, once again, to ensure it is within the saturation limits.

Unstable Q

System output is simulated as being unstable. Not implemented to protect magnetic bearing from rattling and damage.

Non-Robust Stable

![Graphs showing implementation output and control](image1)

Figure 9: Repetitive Control Implemented Reference Tracking for Translation System (left) and Controller (right) Output

![Graphs showing tracking error](image2)

Figure 10: Repetitive Control Implemented Tracking Error Entire Trial (left) and Zoomed-in (right)

Note the controller’s ability to track a reference, which contains a number of frequencies, almost exactly. Tracking is worse at the corners where higher frequency content is involved, exactly where the controller’s tracking performance reaches its limit.
In examining the tracking error plots, we observe the tracking improving over time as the controller learns the reference signal, and works to reduce the error between the reference and output. It does so by accumulating the error in between time intervals and adding it to the output, and thus approaches the reference.

Robustly Stable Q

Like the previous design case, observe the accuracy of tracking the triangle wave. The control output is also steady and well within the saturation bound as desired.

Note that the tracking error in the translation decreases over time as the system learns to track the reference. And the error in rotation remains close to zero as expected since no input is given.
Summary

The repetitive controller achieves reference tracking to multiple frequencies, a feat the other controllers in this report did not handle. By using the IMP for any periodic signal, we were able to insert this dynamic into the controller.

Although we can insert such periodic dynamics into the RST and state-observer controllers, doing so may prove disastrous in implementation. For periodic signals of high orders, numerical issues arise such as slower computing times, high computation errors, and requires for high-order filters. Repetitive control, on the other hand, effectively handles large orders with ease.

This controller is ideal in learning control applications where the reference signal is periodic and known.
# APPENDIX

<table>
<thead>
<tr>
<th>MATLAB FILE</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>system_id</td>
<td>Performs analytical modeling and experimental validation, plant decoupling and uncertainty</td>
</tr>
<tr>
<td>lead_lag_design</td>
<td>Step responses, Nyquist and Bode Plots</td>
</tr>
<tr>
<td>state_feedback_observer_full</td>
<td>State Estimator Feedback</td>
</tr>
<tr>
<td>state_feedback_observer_integrator_full</td>
<td>Augmented with integrator</td>
</tr>
<tr>
<td>modelmatch</td>
<td>RST Controller</td>
</tr>
<tr>
<td>dioph</td>
<td>Solves the Diophantine Equation</td>
</tr>
<tr>
<td>RST</td>
<td>Solves for R, S, and T via the Diophantine Equation</td>
</tr>
<tr>
<td>Youla_example1</td>
<td>H2 and H-inf controller, sensitivity and robust stability, simulation</td>
</tr>
<tr>
<td>H2ModelMatching</td>
<td>Solves the H2 and H-inf controller via model matching</td>
</tr>
<tr>
<td>designZeroPhase</td>
<td>Zero-phase Error Tracking Controller</td>
</tr>
<tr>
<td>zeroPhase</td>
<td>Returns transfer function of FF block resulting in linear phase system</td>
</tr>
<tr>
<td>designRepControl</td>
<td>Repetitive Control</td>
</tr>
</tbody>
</table>
%% MAE 277 System ID Written by David Luong, January 2008
clc;
clear;
close all;

%% Load DSA Data Structures
load TYR
load TUR
load f

%% Extract pertinent data from actual data
to get only the X direction
TUR_X = TUR([1,3],[1,3],:);
TYR_X = TYR([1,3],[1,3],:);
to get only the Y direction
TUR_Y = TUR([2,4],[2,4],:);
TYR_Y = TYR([2,4],[2,4],:);

Decoupling Transformation
T = [1 1;1 -1]/sqrt(2);

Generate Original and Decoupled Plant, G and Gd, for X and Y
for i=1:1601
    G_X(:,:,i) = TYR_X(:,:,i)*inv(TUR_X(:,:,i));
    Gd_X(:,:,i) = T*G_X(:,:,i)*inv(T);
    G_Y(:,:,i) = TYR_Y(:,:,i)*inv(TUR_Y(:,:,i));
    Gd_Y(:,:,i) = T*G_Y(:,:,i)*inv(T);
end

%%
Original Plant Frequency Response in X
% [Gx11 Gx12]
% [Gx21 Gx22]

figure(1)
subplot(2,2,1)
plot(log(f),db(abs(squeeze(G_X(1,1,:)))))
title('Gx11')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,2)
plot(log(f),db(abs(squeeze(G_X(1,2,:)))))
title('Gx12')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,3)
plot(log(f),db(abs(squeeze(G_X(2,1,:)))))
title('Gx21')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,4)
plot(log(f),db(abs(squeeze(G_X(2,2,:)))))
title('Gx22')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

% Original Plant Frequency Response in Y
% \[ \begin{bmatrix} Gy11 & Gy12 \\ Gy21 & Gy22 \end{bmatrix} \]
figure(2)
subplot(2,2,1)
plot(log(f),db(abs(squeeze(G_Y(1,1,:)))))
title('Gy11')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,2)
plot(log(f),db(abs(squeeze(G_Y(1,2,:)))))
title('Gy12')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,3)
plot(log(f),db(abs(squeeze(G_Y(2,1,:)))))
title('Gy21')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,4)
plot(log(f),db(abs(squeeze(G_Y(2,2,:)))))
title('Gy22')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

% Decoupled Plant Frequency Response in X
% \[ \begin{bmatrix} Gdx11 & Gdx12 \\ Gdx21 & Gdx22 \end{bmatrix} \]
figure(3)
subplot(2,2,1)
plot(log(f),db(abs(squeeze(Gd_X(1,1,:)))))
title('Gdx11')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,2)
plot(log(f),db(abs(squeeze(Gd_X(1,2,:)))))
title('Gdx12')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,3)
Luong

plot(log(f),db(abs(squeeze(Gd_X(2,1,:)))))
title('Gdx21')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,4)
plot(log(f),db(abs(squeeze(Gd_X(2,2,:)))))
title('Gdx22')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

%%
%Decoupled Plant Frequency Response in Y
%       [Gdy11 Gdy12]
%Gdy=   [           ]
%       [Gdy21 Gdy22]

figure(4)
subplot(2,2,1)
plot(log(f),db(abs(squeeze(Gd_Y(1,1,:)))))
title('Gdy11')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,2)
plot(log(f),db(abs(squeeze(Gd_Y(1,2,:)))))
title('Gdy12')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,3)
plot(log(f),db(abs(squeeze(Gd_Y(2,1,:)))))
title('Gdy21')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

subplot(2,2,4)
plot(log(f),db(abs(squeeze(Gd_Y(2,2,:)))))
title('Gdy22')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

%%
%Store the data we want to fit
Gyd11 = squeeze(Gd_Y(1,1,:));
Gyd22 = squeeze(Gd_Y(2,2,:));

Gxd11 = squeeze(Gd_X(1,1,:));
Gxd22 = squeeze(Gd_X(2,2,:));

%System ID of X and Y in translation (Gd11) and rotation (Gd22)

%Weighting function for curve fit
Wt = (f<20)*1.5 + (f<1000)*.5+1e-9;
% Analog filter least squares fit to frequency response data
[Gyd11num, Gyd11den] = invfreqs(Gyd11, f*2*pi, 2, 3, Wt)
Gyd11_fit_freq = freqs(Gyd11num, Gyd11den, 2*pi*f)

[Gyd22num, Gyd22den] = invfreqs(Gyd22, f*2*pi, 2, 3, Wt)
Gyd22_fit_freq = freqs(Gyd22num, Gyd22den, 2*pi*f)

[Gxd11num, Gxd11den] = invfreqs(Gxd11, f*2*pi, 2, 3, Wt)
Gxd11_fit_freq = freqs(Gxd11num, Gxd11den, 2*pi*f)

[Gxd22num, Gxd22den] = invfreqs(Gxd22, f*2*pi, 2, 3, Wt)
Gxd22_fit_freq = freqs(Gxd22num, Gxd22den, 2*pi*f)

% Fitted Transfer Function
Gyd11_fit = tf(Gyd11num, Gyd11den)
Gyd22_fit = tf(Gyd22num, Gyd22den)
Gxd11_fit = tf(Gxd11num, Gxd11den)
Gxd22_fit = tf(Gxd22num, Gxd22den)

% Plot Fit and Raw Data
figure(5)
loglog(f, abs(Gyd11), f, abs(Gyd11_fit_freq))
xlabel('frequency'), ylabel('magnitude'), title('Gyd11');
legend('raw', 'fit')

figure(6)
loglog(f, abs(Gyd22), f, abs(Gyd22_fit_freq))
xlabel('frequency'), ylabel('magnitude'), title('Gyd22');
legend('raw', 'fit')

figure(7)
loglog(f, abs(Gxd11), f, abs(Gxd11_fit_freq))

figure(8)
loglog(f, abs(Gxd22), f, abs(Gxd22_fit_freq))

save Gyd_fit Gyd11_fit Gyd22_fit
%
% Analytical Model in Y

% define parameters
L = .146;
l = .024;
l2 = .0028;
I0 = 2.53e-4;
m = .1427;
d = 0.0;

A = [0 1 0 0 0 0;
     8750/m 0 0 0 3.5/m 3.5/m;
     0 0 1 0 0;
     0 0 8750*(L/2-l-d)^2)/I0 0 -3.5/I0*(L/2-l-d) 3.5/I0*(L/2-l-d);]
\[
B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
.25/2.2e-4 & 0 \\
0 & .25/2.2e-4 \\
\end{bmatrix};
\]
\[
C = 12000 \begin{bmatrix}
1 & 0 & -(L/2-l2-d) & 0 & 0 & 0 \\
1 & 0 & (L/2-l2+d) & 0 & 0 & 0 \\
\end{bmatrix};
\]
\[
D = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
\end{bmatrix};
\]
\[
Gy_a = \text{ss}(A,B,C,D);
\]
\[
T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 \\
\end{bmatrix};
\]
\[
\text{Ad} = T*A*\text{inv}(T);
\]
\[
\text{Bd} = T*B;
\]
\[
\text{Cd} = C*\text{inv}(T);
\]
\[
\text{Gyd}_a = \text{ss}(\text{Ad},\text{Bd},\text{Cd},D);
\]
\%
\[
\text{2nd tranformation to decouple input/output}
\]
\[
\text{Gyd}_a.b = \text{Gyd}_a.b*\text{inv}([1 1; 1 -1]);
\]
\[
\text{Gyd}_a.c = [1 1;1 -1]*\text{Gyd}_a.c;
\]
\%
\[
coupled system, transfer functions
\]
\%
\[
\text{figure}(9), q = \text{tf}(\text{Gy}_a)
\]
\[
[\text{mag11},\text{phase11},\text{w11}] = \text{bode}(q(1,1));
\]
\[
[\text{mag12},\text{phase12},\text{w12}] = \text{bode}(q(1,2));
\]
\[
[\text{mag21},\text{phase21},\text{w21}] = \text{bode}(q(2,1));
\]
\[
[\text{mag22},\text{phase22},\text{w22}] = \text{bode}(q(2,2));
\]
\%
\[
\text{subplot}(2,2,1)
\]
\[
\text{semilogx}(\text{w11},\text{squeeze}(\text{db}(\text{mag11})))
\]
\[
\text{axis}([1e2 1e5 -150 50])
\]
\xilabel('frequency'),\yilabel('magnitude')
\[
\text{title('G11')}
\]
\%
\[
\text{subplot}(2,2,2)
\]
\[
\text{semilogx}(\text{w12},\text{squeeze}(\text{db}(\text{mag12})))
\]
\[
\text{axis}([1e2 1e5 -150 50])
\]
\xilabel('frequency'),\yilabel('magnitude')
\[
\text{title('G12')}
\]
\%
\[
\text{subplot}(2,2,3)
\]
\[
\text{semilogx}(\text{w21},\text{squeeze}(\text{db}(\text{mag21})))
\]
\[
\text{axis}([1e2 1e5 -150 50])
\]
\xilabel('frequency'),\yilabel('magnitude')
title('G21')
subplot(2,2,4)
semilogx(w22,squeeze(db(mag22)))
axis([1e2 1e5 -150 50])
xlabel('frequency'),ylabel('magnitude')
title('G22')

% decoupled system, transfer functions
figure(10), qd = tf(Gyd_a)
[magd11, phased11, wd11] = bode(qd(1,1));
[magd12, phased12, wd12] = bode(qd(1,2));
[magd21, phased21, wd21] = bode(qd(2,1));
[magd22, phased22, wd22] = bode(qd(2,2));

subplot(2,2,1)
semilogx(wd11,squeeze(db(magd11)))
axis([1e2 1e5 -150 50])
xlabel('frequency'),ylabel('magnitude')
title('Gd11')
 subplot(2,2,2)
semilogx(wd12,squeeze(db(magd12)))
axis([1e2 1e5 -150 50])
xlabel('frequency'),ylabel('magnitude')
title('Gd12')
 subplot(2,2,3)
semilogx(wd21,squeeze(db(magd21)))
axis([1e2 1e5 -150 50])
xlabel('frequency'),ylabel('magnitude')
title('Gd21')
 subplot(2,2,4)
semilogx(wd22,squeeze(db(magd22)))
axis([1e2 1e5 -150 50])
xlabel('frequency'),ylabel('magnitude')
title('Gd22')

Gyd11_a = tf(Gyd_a(1,1));
Gyd22_a = tf(Gyd_a(2,2));

save Gyd_a Gyd11_a Gyd22_a

%%
% Plot Fit, Raw Data, Analytical Model
figure(11)
subplot(2,1,1)
[mag11, pha11] = bode(Gyd_a(1,1), 2*pi*f);
loglog(2*pi*f, abs(Gyd11), 2*pi*f, abs(Gyd11_fit_freq), 2*pi*f,
abs(squeeze(mag11)))
title('System ID of Gyd11')
xlabel('rad/s')
ylabel('Magnitude')
legend('Experimental', 'Curve Fit', 'Analytical')

subplot(2,1,2)
semilogx(2*pi*f, 180/pi*angle(Gyd11), 2*pi*f,
unwrap(180/pi*angle(Gyd11_fit_freq))+360, 2*pi*f,
squeeze(pha11))
xlabel('rad/s')
ylabel('Phase [rad]')
```matlab
figure(12)
subplot(2,1,1)
[mag22,pha22] = bode(Gyd_a(2,2),2*pi*f);
loglog(2*pi*f, abs(Gyd22), 2*pi*f, abs(Gyd22_fit_freq), 2*pi*f,
abs(squeeze(mag22))
title('System ID of Gyd22')
xlabel('rad/s')
ylabel('Magnitude')
legend('Experimental','Curve Fit','Analytical')

subplot(2,1,2)
semilogx(2*pi*f, 180/pi*angle(Gyd22), 2*pi*f,
180/pi*unwrap(angle(Gyd22_fit_freq)), 2*pi*f, (squeeze(pha22)))
xlabel('rad/s')
ylabel('Phase [rad]')

%% Wr
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd11
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd22

%Weighting function for curve fit
Wt = (f<20)*20.5 + (f<2000)*.5+1e-9;

Wr1_tf = zpk([-2000 -800],[-1e4],.5e-3);%zpk([25000 1000 1000 1000
1000 900 500],1e-8);
Wr1 = freqresp(Wr1_tf,f*2*pi);
Wr2_tf = zpk([-5000 -1000],[-2e4],4e-4);%zpk([3300 1200 1100 1000
1000 500 10],.25e-7);
Wr2 = freqresp(Wr2_tf,f*2*pi);

H1 = freqresp(sysGd11,f*2*pi);
H2 = freqresp(sysGd22,f*2*pi);

delta1 = (squeeze(H1) - Gyd11)./Gyd11;
delta2 = (squeeze(H2) - Gyd22)./Gyd22;

%Analog filter least squares fit to frequency response data
%[Wr11num,Wr11den] = invfreqs(delta11,f*2*pi,15,10,Wt);
%Wr11 = freqs(10*Wr11num,Wr11den,2*pi*f);

% [Wr22num,Wr22den] = invfreqs(delta22,f*2*pi,2,3,Wt);
% Wr22 = freqs(Wr22num,Wr22den,2*pi*f);

%Plot Fit and Raw Data
figure(13)
semilogx(f, db(abs(delta1)), f, db(abs(squeeze(Wr1))))
xlabel('frequency')
ylabel('magnitude')
legend('delta1','Wr1')
title('Wr1')

figure(14)
```

semilogx(f, db(abs(delta2)), f, db(abs(squeeze(Wr2))))
xlabel('frequency')
ylabel('magnitude')
legend('delta1','Wr1')
title('Wr2')

%save Wr
save Wr Wr1_tf Wr2_tf
%% Youla Parameterization of All Stabilizing Controllers
close all;

Q1 = -1./((squeeze(H1) - Gyd11));
Q2 = -1./((squeeze(H2) - Gyd22));

%Plot Fit and Raw Data
figure(1)
semilogx(f, db(Q1))
xlabel('frequency')
ylabel('magnitude')
legend('Q1 Data','Q1 Fit')
title('Q1')

figure(2)
semilogx(f, db(Q2))
xlabel('frequency')
ylabel('magnitude')
legend('Q2 Data','Q2 Fit')
title('Q2')
%% MAE 277 Lead-lag Controller Design
%% David Luong, Winter 2008
%
cic;
clear;
close all;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd11
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd22
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Wr

Ts = 1/5000;
Wr1 = c2d(Wr1_tf,Ts,'tustin');
Wr2 = c2d(Wr2_tf,Ts,'tustin');

%% Lead-Lag Indirect Sisotool Design
%% Lead-lag Controller Sisotool Indirect Design

close all;

Gyd11_s = sysGd11;
Gyd22_s = sysGd22;
Cy11_s = 6.2829*tf([.0028*.086 .0028+.086 1],[.0018 1 0]);
Cy22_s = Cy11_s;
L11_s = Cy11_s*Gyd11_s;
L22_s = Cy22_s*Gyd22_s;

%Controller Implementation
%Step 1: (G(s),C(s)) is stable?
figure(1)
rlocus(L11_s)
figure(2)
rlocus(L22_s)
figure(3)
bode(L11_s)
figure(4)
bode(L22_s)
figure(5)
step(feedback(L11_s,1))
figure(6)
step(feedback(L22_s,1))

%Step 2: (Gzoh(z),C(z)) stable?
Gyd11_z = c2d(Gyd11_s,Ts,'zoh')
Gyd22_z = c2d(Gyd22_s,Ts,'zoh')
Cy11_z = c2d(Cy11_s,Ts,'tustin');
Cy22_z = Cy11_z;
L11_z = Cy11_z*Gyd11_z;
\[ L_{22} = C_y_{22} \times G_{yd_{22}}; \]

```matlab
figure(7)
riocus(L_{11_z})

figure(8)
riocus(L_{22_z})

figure(9)
bode(L_{11_z}, L_{11_s})

figure(10)
bode(L_{22_z}, L_{22_s})
```

% Define sensitivity and transfer functions

\[ T_{11_z} = feedback(L_{11_z}, 1); \]
\[ S_{11_z} = 1 - T_{11_z}; \]

\[ T_{22_z} = feedback(L_{22_z}, 1); \]
\[ S_{22_z} = 1 - T_{22_z}; \]

```matlab
figure(11)
step(T_{11_z})

figure(12)
bode(C_{y11_z}, C_{y11_s})
legend('Discrete', 'Continuous')
```

% Plot Sensitivity and Complementary Sensitivity Functions

```matlab
figure(13), bode(S_{11_z}, S_{22_z}), grid on
title('Sensitivity'), legend('Translation', 'Rotation')
figure(14), bode(T_{11_z}, T_{22_z}), grid on
title('Complementary Sensitivity'), legend('Translation', 'Rotation')
```

% Robust Stability Test

```matlab
figure(15); bodemag(W_r*T_{11_z}), grid on
title('Robust Stability: (W_r*T)_{translation}')
xlabel('rad/sec'), ylabel('magnitude')
figure(16); bodemag(W_r*T_{22_z}), grid on
title('Robust Stability: (W_r*T)_{rotation}')
xlabel('rad/sec'), ylabel('magnitude')
```

%% Lead-Lag Direct Sisotool Design

close all;

```matlab
figure(1)
riocus(G_{yd11_z})

figure(2)
riocus(G_{yd22_z})
```

% Direct Controller Implementation

% Step 1: (G(z), C(z)) stable?

\[ z = tf('z', T_s); \]
\text{Cz}_{11\text{\_} direct} = 1.7762\times(z-0.9202)\times(z-0.998)/((z-0.7563)\times(z-1));
\text{Cz}_{22\text{\_} direct} = \text{Cz}_{11\text{\_} direct};
\text{L}_{11\text{\_} z\text{\_} direct} = \text{Cz}_{11\text{\_} direct}\times\text{Gyd}_{11\text{\_} z};
\text{L}_{22\text{\_} z\text{\_} direct} = \text{Cz}_{22\text{\_} direct}\times\text{Gyd}_{22\text{\_} z};

\text{figure}(3)
\text{rlocus}(\text{L}_{11\text{\_} z\text{\_} direct})

\text{figure}(4)
\text{rlocus}(\text{L}_{22\text{\_} z\text{\_} direct})

\text{figure}(5)
\text{bode}(\text{L}_{11\text{\_} z\text{\_} direct})

\text{figure}(6)
\text{bode}(\text{L}_{22\text{\_} z\text{\_} direct})

\%\text{Define sensitivity and transfer functions}
\text{T}_{11\text{\_} z\text{\_} direct} = \text{feedback}(\text{L}_{11\text{\_} z\text{\_} direct},1);
\text{S}_{11\text{\_} z\text{\_} direct} = 1 - \text{T}_{11\text{\_} z\text{\_} direct};

\text{T}_{22\text{\_} z\text{\_} direct} = \text{feedback}(\text{L}_{22\text{\_} z\text{\_} direct},1);
\text{S}_{22\text{\_} z\text{\_} direct} = 1 - \text{T}_{22\text{\_} z\text{\_} direct};

\text{figure}(7)
\text{step}(\text{T}_{11\text{\_} z\text{\_} direct})

\text{figure}(8)
\text{step}(\text{T}_{22\text{\_} z\text{\_} direct})

\%\text{Plot Sensitivity and Complementary Sensitivity Functions}
\text{figure}(9), \text{bode}(\text{S}_{11\text{\_} z\text{\_} direct},\text{S}_{22\text{\_} z\text{\_} direct}), \text{grid on}
title('\text{Sensitivity}', legend('Translation', 'Rotation'))
\text{figure}(10), \text{bode}(\text{T}_{11\text{\_} z\text{\_} direct},\text{T}_{22\text{\_} z\text{\_} direct}), \text{grid on}
title('\text{Complementary Sensitivity}', legend('Translation', 'Rotation'))

\%\text{Robust Stability Test}
\text{figure}(11); \text{bodemag}(\text{Wr}_{1}\times\text{T}_{11\text{\_} z\text{\_} direct}), \text{grid on}
title('\text{Robust Stability: (Wr_{T})_{\text{\_}translation}}', xlabel('rad/sec'), ylabel('magnitude'))
\text{figure}(12); \text{bodemag}(\text{Wr}_{2}\times\text{T}_{22\text{\_} z\text{\_} direct}), \text{grid on}
title('\text{Robust Stability: (Wr_{T})_{\text{\_}rotation}}', xlabel('rad/sec'), ylabel('magnitude'))

\%\% \text{Simulation of Indirect Controller}
\text{close all;}
\%\text{Insert system parameters}
\text{load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match}
\text{T} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}/\sqrt{2};
\text{Ts} = 1/5000;
\%\text{Insert discretized controller here}
Cy_translation = Cy11_z;
Cy_rotation = Cy11_z;

sim('MagBearing_lead_lag_simulation',2)

figure(1); plot(t,r(:,1), 'b', t,y(:,1), 'r');
title('Simulation Output: Y_{translation}');
xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2), 'b', t,y(:,2), 'r');
title('Simulation Output: Y_{rotation}');
xlabel('time'), ylabel('volts')

figure(3); plot(t,r(:,1), 'b', t,u(:,1), 'r');
title('Simulation Control: U_{translation}');
xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2), 'b', t,u(:,2), 'r');
title('Simulation Control: U_{rotation}');
xlabel('time'), ylabel('volts')

%% Implementation of Indirect Controller

close all;

load lead_lag_compensator_sim_indirect_y_dluong1
tsim = t;
rsim = r;
ysim = y;
usim = u;

load lead_lag_compensator_exp_indirect_y_dluong1
texp = t;
rexp = r;
yexp = y;
uexp = u;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match

figure(1);
plot(tsim,rsim(:,1), 'b', tsim,ysim(:,1), 'r', texp,yexp(:,1), 'k');
title('Implementation and Simulation Output: Y_{translation}');
xlabel('time'), ylabel('volts')
figure(2);
plot(tsim,rsim(:,2), 'b', tsim,ysim(:,2), 'r', texp,yexp(:,2), 'k');
title('Implementation and Simulation Output: Y_{rotation}');
xlabel('time'), ylabel('volts')

figure(3);
plot(tsim,rsim(:,1), 'b', tsim,usim(:,1), 'r', texp,uexp(:,1), 'k');
title('Implementation and Simulation Control: U_{translation}');
xlabel('time'), ylabel('volts')
figure(4);
plot(tsim,rsim(:,2), 'b', tsim,usim(:,2), 'r', texp,uexp(:,2), 'k');
title('Implementation and Simulation Control: U_{rotation}');
xlabel('time'), ylabel('volts')
%% Simulation of Direct Controller

close all;

% Insert system parameters
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match
T = [1 1; 1 -1]/sqrt(2);
Ts = 1/5000;

% Insert discretized controller here
Cy_translation = Cz11_direct;
Cy_rotation = Cz11_direct;

sim('MagBearing_lead_lag_simulation',2)

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Simulation Output: Y_{translation}')
xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Simulation Output: Y_{rotation}')
xlabel('time'), ylabel('volts')

figure(3); plot(t,r(:,1),'b',t,u(:,1),'r');
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r');

%% Implementation of Direct Controller

load lead_lag_compensator_sim_direct_y_dluong1
tsim = t;
rsim = r;
ysim = y;
usim = u;

load lead_lag_compensator_exp_direct_y_dluong1
texp = t;
rexp = r;
yexp = y;
uxexp = u;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match

figure(1);
plot(tsim,rsim(:,1),'b',tsim,ysim(:,1),'r',texp,yexp(:,1),'k');
title('Implementation and Simulation Output: Y_{translation}')
xlabel('time'), ylabel('volts')
figure(2);
plot(tsim,rsim(:,2),'b',tsim,ysim(:,2),'r',texp,yexp(:,2),'k');
title('Implementation and Simulation Output: Y_{rotation}')
xlabel('time'), ylabel('volts')

figure(3);
plot(tsim,rsim(:,1),'b',tsim,usim(:,1),'r',texp,uexp(:,1),'k');
title('Implementation and Simulation Control: U_{translation}')
xlabel('time'), ylabel('volts')
figure(4);
plot(tsim,rsim(:,2),'b',tsim,usim(:,2),'r',texp,uexp(:,2),'k');
title('Implementation and Simulation Control: U_{rotation}')
xlabel('time'), ylabel('volts')
clc;
clear;
close all;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_a
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_fit
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd11
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd22
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Wr

T = [1 1; 1 -1]/sqrt(2);
Ts = 1/5000;

Wr1 = c2d(Wr1_tf,Ts,'tustin');
Wr2 = c2d(Wr2_tf,Ts,'tustin');

%Enter continuous-time Gyd
Gyd11_s = ss(sysGd11);
Gyd22_s = ss(sysGd22);

%Discrete-time Gyd11
Gyd11_z = c2d(Gyd11_s,Ts,'tustin');
A1 = Gyd11_z.a;
B1 = Gyd11_z.b;
C1 = Gyd11_z.c;
D1 = Gyd11_z.d;

%Discrete-time Gyd22
Gyd22_z = c2d(Gyd22_s,Ts,'tustin');
A2 = Gyd22_z.a;
B2 = Gyd22_z.b;
C2 = Gyd22_z.c;
D2 = Gyd22_z.d;

Psf1 = [.95 .83 .84];
Pobs1 = [.01 .4 .02];

Psf2 = Psf1;
Pobs2 = Pobs1;

K1 = place(A1,B1,Psf1);
L1 = place(A1','C1','Pobs1);
L1 = L1';

K2 = place(A2,B2,Psf2);
L2 = place(A2','C2','Pobs2);
L2 = L2';

N1 = inv(C1*inv(eye(length(A1))-A1+B1*K1)*B1);
N2 = inv(C2*inv(eye(length(A2))-A2+B2*K2)*B2);
%% Sensitivity and Complementary Sensitivity

[nGd1,dGd1] = ss2tf(A1,B1,C1,0);
Gd1 = tf(nGd1,dGd1,Ts);
[num1,den1] = ss2tf(A1-L1*C1-B1*K1,L1,K1,0);
factor1 = tf(num1,den1,Ts);
LG1 = factor1*Gd1;
Sy1 = 1/(1+LG1);
Ty1 = 1-Sy1;

[nGd2,dGd2] = ss2tf(A2,B2,C2,0);
Gd2 = tf(nGd2,dGd2,Ts);
[num2,den2] = ss2tf(A2-L2*C2-B2*K2,L2,K2,0);
factor2 = tf(num2,den2,Ts);
LG2 = factor2*Gd2;
Sy2 = 1/(1+LG2);
Ty2 = 1-Sy2;

%Plot Sensitivity and Complementary Sensitivity Functions
figure(1), bode(Sy1,Sy2), grid on
title('Sensitivity'), legend('Translation', 'Rotation')
figure(2), bode(Ty1,Ty2), grid on
title('Complementary Sensitivity'), legend('Translation', 'Rotation')

%% Robust Stability

close all;

figure(1); bodemag(Wr1*Ty1), grid on
title('Robust Stability: (Wr*T)_{translation}')
xlabel('rad/sec'), ylabel('magnitude')
figure(2); bodemag(Wr2*Ty2), grid on
title('Robust Stability: (Wr*T)_{rotation}')
xlabel('rad/sec'), ylabel('magnitude')

%% Simulation

sim('MagBearing_SF_obs_simulation',2);
%sim('MagBearing_state_feedback',2);
figure(1)
plot(t,y,t,r)
%% MAE 277 Discrete-time State Estimation Feedback with Integrator  
% Augmented with Integrator
% David Luong, Feb 2008
%
clc;
clear;
close all;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_a
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_fit
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd11
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd22
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Wr

T = [1 1; 1 -1]/sqrt(2);
Ts = 1/5000;
Wr1 = c2d(Wr1_tf,Ts,'tustin');
Wr2 = c2d(Wr2_tf,Ts,'tustin');

%Enter continuous-time Gyd
Gyd11_s = ss(sysGd11);
Gyd22_s = ss(sysGd22);

%Discrete-time Gyd11
Gyd11_z = c2d(Gyd11_s,Ts,'tustin');
A1 = Gyd11_z.a;
B1 = Gyd11_z.b;
C1 = Gyd11_z.c;
D1 = Gyd11_z.d;

%Discrete-time Gyd22
Gyd22_z = c2d(Gyd22_s,Ts,'tustin');
A2 = Gyd22_z.a;
B2 = Gyd22_z.b;
C2 = Gyd22_z.c;
D2 = Gyd22_z.d;

%Desired closed-loop pole locations
Psf1 = [ .8 .83 .84 .9];
Pobs1 = [.01 .4 .02];
Psf2 = Psf1;
Pobs2 = Pobs1;

%Find Augmented Matrices for integrator
Aaug1 = [A1 zeros(3,1); C1 1];
Baug1 = [B1;1];
Aaug2 = [A2 zeros(3,1); C2 1];
Baug2 = [B2;1];

%Find state feedback and observer gains
K1 = place(Aaug1,Baug1,Psf1);
K1 = K1(4);
K1 = K1(1:3);
L1 = place(A1',C1',Pobs1);
L1 = L1';
K2 = place(Aaug2,Baug2,Psf2);
Ki2 = K2(4);
K2 = K2(1:3);
L2 = place(A2',C2',Pobs2);
L2 = L2';

%% Sensitivity and Complementary Sensitivity Functions
close all;
z = tf([1 0],1,Ts);

%Define state-space of integrator internal model
Kd1 = Ki1;
Ad1 = 1;
Bd1 = 1;
int1 = tf([Ki1 0],[1 -1],Ts);

Kd2 = Ki2;
Ad2 = 1;
Bd2 = 1;
int2 = tf([Ki2 0],[1 -1],Ts);

[nGd1,dGd1] = ss2tf(A1,B1,C1,0);
Gd1 = tf(nGd1,dGd1,Ts);
[nOB11,dOB11] = ss2tf(A1-L1*C1,B1,K1,0);
OB11 = tf(nOB11,dOB11,Ts);
[nOB12,dOB12] = ss2tf(A1-L1*C1,L1,K1,0);
OB12 = tf(nOB12,dOB12,Ts);
LG1 = 1/(1+OB11)*Gd1*(OB12 + int1);
Sy1 = 1/(1+LG1);
Ty1 = 1-Sy1;

[nGd2,dGd2] = ss2tf(A2,B2,C2,0);
Gd2 = tf(nGd2,dGd2,Ts);
[nOB21,dOB21] = ss2tf(A2-L2*C2,B2,K2,0);
OB21 = tf(nOB21,dOB21,Ts);
[nOB22,dOB22] = ss2tf(A2-L2*C2,L2,K2,0);
OB22 = tf(nOB22,dOB22,Ts);
LG2 = 1/(1+OB21)*Gd2*(OB22 + int2);
Sy2 = 1/(1+LG2);
Ty2 = 1-Sy2;

%Plot Sensitivity and Complementary Sensitivity Functions
figure(1), bode(Sy1,Sy2), grid on
title('Sensitivity'), legend('Translation', 'Rotation')
figure(2), bode(Ty1,Ty2), grid on
title('Complementary Sensitivity'), legend('Translation', 'Rotation')

%% Robust Stability
close all;

figure(1); bodemag(Wr1*Ty1), grid on
title('Robust Stability: (Wr*T)_{translation}')
xlabel('rad/sec'), ylabel('magnitude')
figure(2); bodemag(Wr2*Ty2), grid on
title('Robust Stability: (Wr*T)_{rotation}')
xlabel('rad/sec'), ylabel('magnitude')
%% Simulation
sim('MagBearing_SF_obs_integrator_simulation',2);

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Simulation Output: Y_{translation}
xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Simulation Output: Y_{rotation}
xlabel('time'), ylabel('volts')
close all
clear all
clc;

load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_a
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Gyd_fit
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd11
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\sysGd22
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\G_y_10Order_96match
load G:\Documents\UCLA\MAE277\DSAdataWithDoc\Wr

Ts = 1/5000;
s = tf('s');
z = tf('z', Ts);
Wr1 = c2d(Wr1_tf,Ts,'tustin');
Wr2 = c2d(Wr2_tf,Ts,'tustin');

%% RST on 3rd Order Models, Simulation on 10th
% Discretize plants
Gt_d = c2d(sysGd11, Ts, 'zoh');
Gt_d = zpk(Gt_d);
Gr_d = c2d(sysGd22, Ts, 'zoh');
Gr_d = zpk(Gr_d);

% Identify RST variables
Bu = [1, -1.399]; % (z-1.399);
Bc = [-0.088641, 0.06734];
Ac = [1, -2.446, 1.885, -0.4428]; % (z-1.081)*(z-0.9199)*(z-0.4453);
Ao = [1, -1.95, 1.265, -0.273];
D = [1, -1]; %Integrator
E = 1;
Ampnew = [1, -1.89, 0.893];
% Amp = [1, -1.89, 0.8959];
Bmp = (sum(Ampnew)/sum(Bu));
[Rt,St,Tt] = RST(Au,Ac,Bu,Bc,Ampnew,Bmp,D,E,Ao)
Bur = [1, -1.707];
Bcr = [-0.06933, 0.0503];
Acr = [1, -2.453, 1.893, -0.4442];
Bmpr = (sum(Ampnew)/sum(Bur));
% Amp = [1, -1.89, 0.8959];
[RR,SR,TR] = RST(Au,Acr,Bur,Bcr,Ampnew,Bmpr,D,E,Ao)

% sensitivity functions
Bt = [-0.088641, 0.1913, -0.09421];
Br = [-0.06933, 0.1687, -0.08586];

% Complementary Sensitivity Functions
T_regt =
tf(conv(Bu,Bc),1,Ts)*tf(Tt,1,Ts)/(tf(conv(Ac,Rt),1,Ts)+tf(conv(conv(Bu,Bc),St),1,Ts));
T_regr =
tf(conv(Bur,Bcr),1,Ts)*tf(Tr,1,Ts)/(tf(conv(Acr,RR),1,Ts)+tf(conv(conv(Bur,Bcr),Sr),1,Ts));
% Complementary Sensitivity Functions
S_regt = 1 - T_regt;
S_regr = 1 - T_regr;

% set pamemters for simulation;
T = [1 1; 1 -1]/ sqrt(2); % transformation matrix

% put your controller here;
Rt = tf([1 0 0 0],Rt, Ts);
check3 = zpk(Rt)
St = tf(St, [1 0 0 0], Ts);
Tt = tf(Tt, [1 0 0 0], Ts);
Rr = tf([1 0 0 0],Rr, Ts);
check4 = zpk(Rr)
Sr = tf(Sr, [1 0 0 0], Ts);
Tr = tf(Tr, [1 0 0 0], Ts);

%% Simulation
% run the simulation on 10th Order System;
sim('RST_simulation',1);

%% plot figures;
figure(1); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Simulation Output: Y_{translation}'); xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Simulation Output: Y_{rotation}'); xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'r');
title('Simulation Control: U_{translation}'); xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r');
title('Simulation Control: U_{rotation}'); xlabel('time'), ylabel('volts')

%% Sensitivity and Complementary Sensitivity Functions
figure(5), bode(S_regt,S_regr), grid on
title('Sensitivity'); legend('Translation', 'Rotation')
figure(6), bode(T_regt,T_regr), grid on
title('Complementary Sensitivity'); legend('Translation', 'Rotation')

%% Robust Stability Test
figure(7); bodemag(Wr1*T_regt), grid on
title('Robust Stability: (Wr*T)_{translation}'); xlabel('rad/sec'), ylabel('magnitude')
figure(8); bodemag(Wr2*T_regr), grid on
title('Robust Stability: (Wr*T)_{rotation}'); xlabel('rad/sec'), ylabel('magnitude')

%% RST on 3rd Order Models, Simulation on 10th, Sinusoidal Tracking
% Discretize plants
Gt_d = c2d(sysGd11, Ts, 'zoh');
Gr_d = c2d(sysGd22, Ts, 'zoh');

% Identify RST variables
Bu = [1, -1.399]; % (z-1.399);
Bc = [-0.088641, 0.06734];
\[ Au = 1; \]
\[ Ac = [1, -2.446, 1.885, -0.4428]; \% (z-1.081)*(z-0.9199)*(z-0.4453); \]
\[ Ao = [1, -1.95, 1.265, -0.273]; Ao_new = conv(Ao,[1 -.95]); Ao = Ao_new; \]
\[ w_0 = 15*2*pi; \]
\[ D = [1, -2*cos(w_0*Ts), 1]; \% Sinusoidal Rejector \]
\[ E = 1; \]
\[ Amp_new = [1, -1.89, 0.893]; \% Amp = [1, -1.89, 0.8959]; \]
\[ Bmp = (sum(Amp_new)/sum(Bu)); \]
\[ [Rt,St,Tt] = RST(Au,Ac,Bu,Bc,Amp_new,Bmp,D,E,Ao) \]
\[ Bur = [1, -1.707]; \]
\[ Bcr = [-0.06933, 0.0503]; \]
\[ Acr = [1, -2.453, 1.893, -0.4442]; \]
\[ Bmpr = (sum(Amp_new)/sum(Bur)); \% Amp = [1, -1.89, 0.8959]; \]
\[ [Rr,Sr,Tr] = RST(Au,Acr,Bur,Bcr,Amp_new,Bmpr,D,E,Ao) \%
\% sensitivity functions???
\[ Bt = [-0.088641, 0.1913, -0.09421]; \]
\[ Br = [-0.06933, 0.1687, -0.08586]; \%
\% Sensitivity Functions
\[ S_regt = \]
\[ tf(Ac,1,Ts)*tf(Rt,1,Ts)/(tf(conv(Ac,Rt),1,Ts)+tf(conv(conv(Bu,Bc),St),1,Ts)); \]
\[ S_regr = \]
\[ tf(Ac,1,Ts)*tf(Rr,1,Ts)/(tf(conv(Ac,Rr),1,Ts)+tf(conv(conv(Bur,Bcr),Sr),1,Ts)); \]
\% Complementary Sensitivity Functions
\[ T_regt = 1 - S_regt; \]
\[ T_regr = 1 - S_regr; \%
\% set parameters for simulation;
\[ T = [1 1; 1 -1]/sqrt(2); \% transformation matrix \%
\% Define RST Controller;
\[ Rt = tf(1,Rt,Ts); \]
\[ St = tf(St,1,Ts); \]
\[ Tt = 1; \]
\[ SDRt = St*Rt; \]
\[ St = 1; \]
\[ Rr = tf(1,Rr,Ts); \]
\[ Sr = tf(Sr,1,Ts); \]
\[ Tr = 1; \]
\[ SDRr = Sr*Rr; \]
\[ Sr = 1; \%
\% Simulation
\% run the simulation on 10th Order System;
\[ sim('RST_sin_simulation',1); \]
\% plot figures;
figure(1); plot(t,r(:,1),'b',t,y(:,1),'rx');
title('Simulation Output: Y_{translation}'); xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'rx');
title('Simulation Output: Y_{rotation}'); xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'rx');
title('Simulation Control: U_{translation}'); xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'rx');
title('Simulation Control: U_{rotation}'); xlabel('time'), ylabel('volts')

%% Sensitivity and Complementary Sensitivity Functions
figure(5), bode(S_regt,S_regr), grid on
title('Sensitivity'); legend('Translation', 'Rotation')
figure(6), bode(T_regt,T_regr), grid on
title('Complementary Sensitivity'); legend('Translation', 'Rotation')

%% Robust Stability Test
figure(7); bodemag(Wr1*T_regt), grid on
title('Robust Stability: (Wr*T)_{translation}'); xlabel('rad/sec'), ylabel('magnitude')
figure(8); bodemag(Wr2*T_regr), grid on
title('Robust Stability: (Wr*T)_{rotation}'); xlabel('rad/sec'), ylabel('magnitude')

%% RST Implementation without Integrator

%% RST Implementation with Integrator

close all;
clear all;

%load RST_exp_David % 1st Implementation
load G:\Documents\UCLA\MAE277\Controller_Design\RST_exp_David %2nd Implementation

figure(1); plot(t,r(:,1),'b',t,y(:,1)-.6,'r');
title('Implementation Output: Y_{translation}'); xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2)-.6,'r');
title('Implementation Output: Y_{rotation}'); xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'r');
title('Implementation Control: U_{translation}'); xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r');
title('Implementation Control: U_{rotation}'); xlabel('time'), ylabel('volts')

%% RST Implementation with Integrator

close all;
clear all;
clc;
load RST_exp_int_David

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r'); title('Implementation Output: Y_{translation}'); xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r'); title('Implementation Output: Y_{rotation}'); xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'r'); title('Implementation Control: U_{translation}'); xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r'); title('Implementation Control: U_{rotation}'); xlabel('time'), ylabel('volts')

%% RST Implementation Sinusoidal Reference Tracking

close all;
cic;

load G:\Documents\UCLA\MAE277\Controller_Design\RST_exp_sin_David

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r'); title('Implementation Output: Y_{translation}'); xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r'); title('Implementation Output: Y_{rotation}'); xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'r'); title('Implementation Control: U_{translation}'); xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r'); title('Implementation Control: U_{rotation}'); xlabel('time'), ylabel('volts')
function [X, Y] = Dioph(A,B,C)
% Solve A*X + B*Y = C by forming a Sylvester matrix
% Per Astrom and Whittenmark pp.295

% A: z^n + a1 z^(n-1) + ... + an      [1 a1 a2 ... an] array length: n+1
% B: b0 z^n + b1 z^(n-1) + ... + bm   [b0 b1 ... bm] length: m+1 (m<=n)
% X: y0 z^(p-n) + x1 z^(p-n-1) + ... + xp-n   [x1 x2 ... xp-n] length: p-n
% Y: y0 z^(p-n-1) + ... + yn-1     [y0 y1 ... yn-1] length: n
% Z: z^p + c1 z^(p-1) + ... +cp     [1 c1 c2 ... cp] length: p+1

Y = zeros(1, (length(A)-1));
X = zeros(1, length(C)-length(A));

dim = length(X) + length(Y);
K = zeros(dim, dim);
for j = 1:length(X)
    for i = 1:length(A)
        K(i+j-1,j) = A(i);
    end
end

for j = 1:length(Y)
    for i = 1:length(B)
        K(dim+1-(i+j-1),dim+1-j) = B(length(B)+1-i);
    end
end

K

Cp = zeros(dim,1);
for i = 1:dim
    Cp(i) = C(i+1);
end

for i = 1:length(A)-1
    Cp(i) = Cp(i) - A(i+1);
end

ans = inv(K)*Cp;

for i = 1:length(X)
    X(i) = ans(i);
end

for i = 1:length(Y)
    Y(i) = ans(i+length(X));
end
function [R,S,T] = RST(Au,Ac,Bu,Bc,Amp,Bmp,D,E,Ao)
% I/O model matching and pole placement
% Solve Ac*D*Rpp + Bu*E*Spp = Ao*Amp by forming a Sylverster matrix
% Per Astrom and Whittenmark pp.295 and Tsao's notes
% use [X,Y] = Dioph(A,B,C)

% plant: bp*bn/ap, (bp should be monic, bn should include unstable zeros?)

% to obtain causal controller, the following conditions must be satisfied:
% Feedforward: deg(Amp) - deg(Bmp) >= deg(Ac) - deg(Bc)
% Feedback: deg(Ao) >= 2*deg(A) + deg(D) + deg(E) - deg(Amp) - deg(Au)
%                      deg(Bc) - 1

A = conv(Ac,D);
B = conv(Bu,E);
C = conv(Ao,Amp);
[Rpp,Spp] = Dioph(A,B,C);
R = conv(Bc,conv(D,[1,Rpp]));
S = conv(Au,conv(E,Spp));
T = conv(Bmp,Ao);
% Example for Youla Parameterization and h2/hinf minimization;
% Some comments:
% 1) for unstable plant, we must first find a stabilizing controller 
%    e.g, lead compensator. Then using closing the loop we get the
%    closed-loop system Ghat;
% 2) the sensitivity function is formulated as a function of the free
%    parameter Q, i.e., Wp*S = Wp - Ghat*Gp*Q = t1 - t2*Q;
% 3) the goal is to minimize H_2 and H_inf norm of S;
% By Yigang Wang, 02/28/2007

close all;
clear;
clc;

Fs = 5000;  % sampling rate;
h = 1/Fs;
Ts = h;
omega = logspace(0,5,300);

% load the system models
load sysGd11;
load sysGd22;
load sysWr;
load G_y_10Order_96match;

% plant model;
sysGdz11 = c2d(sysGd11,h,'zoh');
sysGdz22 = c2d(sysGd22,h,'zoh');

% z-domain lead compensator;
cdz11 = 3.5805*tf([1 -0.8394],[1 -0.4817],h);
cdz22 = 3.0817*tf([1 -0.835],[1 -0.485],h);

% closed-loop wr = wr_original * S
wrz1 = minreal(wrz1/(1+sysGdz11*cdz11));
wrz2 = minreal(wrz2/(1+sysGdz22*cdz22));

% stable closed-loop Ghat;
sysGhat11 = feedback(sysGdz11*cdz11,1);
sysGhat22 = feedback(sysGdz22*cdz22,1);

% Wpz : performance weight
%%% Modify this section for better performance %%%%%
wb1=2*pi*80; %cross-over frequency
Ms=10^(5/20); %low frequency gain
ep=10^(-10/20); %high frequency dcgain
wp=makeweight(Ms,wb1,ep);
wpz = c2d(wp,h,'tustin');

% calculate t1, t2;
t1_11 = wpz;
t2_11 = wpz*sysGhat11;
\( t_{1,22} = wpz; \)
\( t_{2,22} = wpz*sysGhat22; \)

% \textbf{H2 minimization;}
\[
[q_{h2\_11} \ h2Gamma_{11}] = H2ModelMatching(t_{1,11},t_{2,11});
[q_{h2\_22} \ h2Gamma_{22}] = H2ModelMatching(t_{1,22},t_{2,22});
\]

%% Sensitivity and Complementary Sensitivity
% calculate sensitivity function;
\[\text{sen}_{h2\_11} = \text{minreal}((t_{1,11}-t_{2,11}*q_{h2\_11})/wpz);\]
\[\text{sen}_{h2\_22} = \text{minreal}((t_{1,22}-t_{2,22}*q_{h2\_22})/wpz);\]
% another way to calculate sensitivity function;
\[\text{sen}_{h2\_11n} = 1-sysGhat11*q_{h2\_11};\]
\[\text{sen}_{h2\_22n} = 1-sysGhat22*q_{h2\_22};\]

% calculate complementary sensitivity function;
\[\text{csen}_{h2\_11n} = sysGhat11*q_{h2\_11};\]
\[\text{csen}_{h2\_22n} = sysGhat22*q_{h2\_22};\]

% performance S calculations;
figure;
\text{subplot}(211); bodemag(sen_{h2\_11}); hold on; bodemag(sen_{h2\_11n});
bodemag(1/wpz); grid on; legend('sensitivity function','sensitivity function','1/W_p',4);
\text{title}('sensitivity function of sysGdz11');
\text{subplot}(212); bodemag(sen_{h2\_22}); hold on; bodemag(sen_{h2\_22n});
bodemag(1/wpz); grid on; legend('sensitivity function','sensitivity function','1/W_p',4);
\text{title}('sensitivity function of sysGdz22');

% performance T calculations;
figure;
\text{subplot}(211); hold on; bodemag(csen_{h2\_11n}); bodemag(1/wpz); grid on;
\text{legend}('complementary sensitivity function','1/W_p',4);
\text{title}('complementary sensitivity function of sysGdz11');
\text{subplot}(212); hold on; bodemag(csen_{h2\_22n}); bodemag(1/wpz); grid on;
\text{legend}('complementary sensitivity function','1/W_p',4);
\text{title}('complementary sensitivity function of sysGdz22');

% Robust Stability
% robust stability TW_{R} = Wr*T = TWR*(1-S)
\text{figure; subplot}(211); bodemag(TWr1); grid on; axis([10 10^5 -50 5]);
\text{title}('robust stability of sysGdz11: |TW_r|');
\text{figure; subplot}(212); bodemag(TWr2); grid on; axis([10 10^5 -50 5]);
\text{title}('robust stability of sysGdz22: |TW_r|');

% High order plant simulation;
% \text{run the simulation;}
\text{sim('MagBearing_x_axis_simulation',2);}
xlabel('time'), ylabel('volts')
figure; plot(t,r(:,2),'b',t,y(:,2),'r');
title('Implementation Output: $Y_{rotation}$')
xlabel('time'), ylabel('volts')
figure; plot(t,r(:,1),'b',t,u(:,1),'r');
title('Implementation Control: $U_{translation}$')
xlabel('time'), ylabel('volts')
figure; plot(t,r(:,2),'b',t,u(:,2),'r');
title('Implementation Control: $U_{rotation}$')
xlabel('time'), ylabel('volts')

%% H2 Implementation

close all;
clear all;
clc;

load data_80_5_N10

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Implementation Output: $Y_{translation}$')
xlabel('time'), ylabel('volts')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Implementation Output: $Y_{rotation}$')
xlabel('time'), ylabel('volts')
figure(3); plot(t,r(:,1),'b',t,u(:,1),'r');
title('Implementation Control: $U_{translation}$')
xlabel('time'), ylabel('volts')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r');
title('Implementation Control: $U_{rotation}$')
xlabel('time'), ylabel('volts')
function [Kz Gamma] = H2ModelMatching(Mz,Pz)
% This function is used to solve H2-norm
% minimization of Model Matching Problem, i.e.
% min ||Mz-KzPz||_2;
%
% Written by Yigang Wang March 3, 2008
%
% Note:
% 1) Mz, Pz must be stable, but they can be any LTI form, i.e.,
% ss,tf..
% 2) Currently, this function is only valid for SISO system;
% Input parameters:
% 1) Mz: reference model in z-domain;
% 2) Pz: plant model in z-domain;
% Output parameters:
% 1) Kz: resulting controller in z-domain;
% 2) Gamma: minimized Hinf norm;
%
% converts everything into the continuous domain
Ts = Mz.ts;
M = d2c(Mz,'tustin');
P = d2c(Pz,'tustin');

% defining the interconnects in the system
systemnames = 'M P';
inputvar = '[r; u]';
outputvar = '[M-P; r]';
input_to_M = '[r]';
input_to_P = '[u]';
T = sysic;

% runs the h2 synthesis toolkit
[K,CL,Gamma]=h2syn(T,1,1);
%
% returns everything back into the discrete domain
Kz = c2d(K,Ts,'tustin');
Gamma = norm(Mz-Pz*Kz,2);

% runs the hinfsyn synthesis toolkit
[K,CL,Gamma]=hinfsyn(T,1,1);
%
% returns everything back into the discrete domain
Kz = c2d(K,Ts,'tustin');
Gamma = norm(Mz-Pz*Kz,inf);
% MAE 277 Zero-Phase Error Tracking Controller
% David Luong, Winter 2008

clear all
close all

load sysGd11
load sysGd22
load G_y_10Order_96match

Ts = 1/5000
T = [1 1; 1 -1]/ sqrt(2)

Gz11 = c2d(sysGd11,Ts,'zoh');
Gz22 = c2d(sysGd22,Ts,'zoh');

Cx1 = zpk([.9174],[.72],2.7163,Ts);
Cx2 = Cx1;

% step(1/(1+Gztrans*Cx1))

Ghat11 = minreal(Gz11*Cx1/(1+Gz11*Cx1))
figure(1),step(Ghat11)
Ghat22 = minreal(Gz22*Cx2/(1+Gz22*Cx2))
figure(2),step(Ghat22)

F11 = zeroPhase(Ghat11)
F22 = zeroPhase(Ghat22)

N1 = 2;
sim('MagBearing_x_axis_simulation_ZeroPhase',.2)

figure(3); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Simulation Output: Y_{translation}'), xlabel('time'), ylabel('volts'), legend('reference','output')
figure(4); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Simulation Output: Y_{rotation}'), xlabel('time'), ylabel('volts'), legend('reference','output')

figure(5); plot(t,r(:,1),'b',t,u(:,1),'r');
title('Simulation Control: U_{translation}'), xlabel('time'), ylabel('volts'), legend('reference','output')
figure(6); plot(t,r(:,2),'b',t,u(:,2),'r');
title('Simulation Control: U_{rotation}'), xlabel('time'), ylabel('volts'), legend('reference','output')

figure(7)
plot(t,r(:,1)-y(:,1))
hold on
plot(t,r(:,2)-y(:,2), 'r')
hold off
title('Simulation Tracking Error'), xlabel('time'), ylabel('volts')
legend('e_{translation}', 'e_{Rotation}')

% Bode Plots of Compensated System
figure(8)
bode(F11*Ghat11)
figure(9)
bode(F22*Ghat22)

%% Implementation
clear all;
cic;
close all;

load zeroPhaseHighFreq %Translation @ 100 Hz, Rotation @ 150 Hz
%load zeroPhaseLowFreq %Translation @ 10 Hz, Rotation @ 20 Hz

figure(1); plot(t,r(:,1), 'b', t,y(:,1), 'r');
title('Implementation Output: Y_{translation}')
xlabel('time'), ylabel('volts')
legend('reference', 'output')
figure(2); plot(t,r(:,2), 'b', t,y(:,2), 'r');
title('Implementation Output: Y_{rotation}')
xlabel('time'), ylabel('volts')
legend('reference', 'output')
figure(3); plot(t,r(:,1), 'b', t,u(:,1), 'r');
title('Implementation Control: U_{translation}')
xlabel('time'), ylabel('volts')
legend('reference', 'output')
figure(4); plot(t,r(:,2), 'b', t,u(:,2), 'r');
title('Implementation Control: U_{rotation}')
xlabel('time'), ylabel('volts')
legend('reference', 'output')

figure(5)
plot(t,r(:,1)-y(:,1))
hold on
plot(t,r(:,2)-y(:,2), 'r')
hold off
title('Implementation Tracking Error'), xlabel('time'), ylabel('volts')
legend('e_{translation}', 'e_{Rotation}')
function xx = zeroPhase(Gsys)
% this function returns a system transfer function
% that when put in series with a stable system
% Gsys will result in a linear phase system
% note: only works on discrete siso systems

[z,p,k,Ts] = zpkdata(Gsys,'v');

% find the cancelable and uncancelable parts
ii = find(abs(z)<1);
jj = find(abs(z)>=1);
stableZ = z(ii);
unstableZ = z(jj);

% invert the invertable parts
Ginv = zpk(p,stabileZ,1/k,Ts);

% take the Bstar transpose
Bstar = flipr(poly(unstableZ));
BstarTF = tf(Bstar,1,Ts);

xx = BstarTF*Ginv/(sum(Bstar)^2);

% make xx realizable
[z,p,k] = zpkdata(xx,'v');
delay = max(length(z)-length(p),0);
xx = xx*zpk([],zeros(1,delay),1,Ts);
clear all
close all
load sysGd11
load sysGd22
load G_y_10Order_96match
Ts = 1/5000
T = [1 1; 1 -1]/ sqrt(2)

Gz11 = c2d(sysGd11,Ts,'zoh');
Gz22 = c2d(sysGd22,Ts,'zoh');

Cx1 = zpk([.9174],[.72],2.7163,Ts);
Cx2 = Cx1;

Ghat11 = minreal(Gz11*Cx1/(1+Gz11*Cx1))
step(Ghat11)
Ghat22 = minreal(Gz22*Cx2/(1+Gz22*Cx2))
step(Ghat22)

F11_prime = zeroPhase(Ghat11)
F22_prime = zeroPhase(Ghat22)

% unstable; Q = 1
% N1 = 2;
% Np = 500;
% gamma = 40;
% q_prime = 1;
% N2 = 0;

% Stabilizing Q ; but not Robustly Stable
N1 = 2;
Np = 500;
gamma = 40;
q_prime = tf([1 2 1],[4 0 0],Ts);
N2 = 1;

% Robust Stable Q
% N1 = 2;
% Np = 500;
% gamma = 40;
% q_prime = tf([1 2 1],[4 0 0],Ts)^40;
% N2 = 40;
% Simulation
close all;
sim('MagBearing_x_axis_simulation_Rep_Control',30)

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r'); title('Simulation Output: Y_{translation}') xlabel('time'), ylabel('volts') legend('reference','output')

figure(2); plot(t,r(:,2),'b',t,y(:,2),'r'); title('Simulation Output: Y_{rotation}') xlabel('time'), ylabel('volts') legend('reference','output')

figure(3); plot(t,r(:,1),'b',t,u(:,1),'r'); title('Simulation Control: U_{translation}') xlabel('time'), ylabel('volts') legend('reference','output')

figure(4); plot(t,r(:,2),'b',t,u(:,2),'r'); title('Simulation Control: U_{rotation}') xlabel('time'), ylabel('volts') legend('reference','output')

figure(5)
plot(t,r(:,1)-y(:,1)) hold on plot(t,r(:,2)-y(:,2),'r') hold off title('Simulation Tracking Error'), xlabel('time'), ylabel('volts') legend('e_{translation}','e_{rotation}')

figure(6)
bode(1/wrk1); hold on; bode(tf([1,2,1],[4 0 0],Ts)^40); bode(tf([1,2,1],[4 0 0],Ts)); legend('1/Wr','Robustly Stable','Stablizing Control')

%% Implementation

load sysWr
wrk2 = wrz2*1/(1+Cx2*Gz22)
wrk1 = wrz1*1/(1+Cx1*Gz11)
figure(6)
bode(1/wrk1); hold on;
bode(tf([1,2,1],[4 0 0],Ts)^40);
bode(tf([1,2,1],[4 0 0],Ts)); legend('1/Wr','Robustly Stable','Stablizing Control')

%% Implementation
clear all;
clc;
close all;

%load repetitiveStable %Translation @ Hz, Rotation @ Hz
load robustStable %Translation @ Hz, Rotation @ Hz

figure(1); plot(t,r(:,1),'b',t,y(:,1),'r');
title('Implementation Output: Y_{translation}')
xlabel('time'), ylabel('volts')
legend('reference','output')
figure(2); plot(t,r(:,2),'b',t,y(:,2),'r');
title('Implementation Output: Y_{rotation}')
xlabel('time'), ylabel('volts')
legend('reference','output')

figure(3); plot(t,r(:,1),'b',t,u(:,1),'r');
title('Implementation Control: U_{translation}')
xlabel('time'), ylabel('volts')
legend('reference','output')
figure(4); plot(t,r(:,2),'b',t,u(:,2),'r');
title('Implementation Control: U_{rotation}')
xlabel('time'), ylabel('volts')
legend('reference','output')

figure(5)
plot(t,r(:,1)-y(:,1))
hold on
plot(t,r(:,2)-y(:,2),'r')
hold off
title('Implementation Tracking Error'), xlabel('time'), ylabel('volts')
legend('e_{translation}','e_{Rotation}')
1 State Observer with Integrator

Figure 1: Block Diagram of State Observer Control Structure

We have the following system and feedback control law:

\[
\begin{align*}
    x(k + 1) &= Ax(k) + Bu(k) \\
    y(k) &= Cx(k) \\
    u(k) &= -Kx(k) + Nr(k)
\end{align*}
\]  

(1)  

(2)  

In order to incorporate the integrator into our design. We need to augment the plant such that we can design for an integrator gain \( K_I \). As a result, we have a augmented plant in which the new gain matrix \( K \) will also calculate the gain \( K_I \) for the integrator. Unfortunately, this cannot be done for the observer gain \( L \), since the augmented system will become unobservable. However, this is not really of concern since the state that is unobservable is the integrator gain that we are calculating ourselves and does not really need to be “estimated”.

\[
\begin{align*}
    \begin{bmatrix}
        \dot{x}(k + 1) \\
        x_I(k + 1)
    \end{bmatrix}
    &=
    \begin{bmatrix}
        A & 0 \\
        C & 1
    \end{bmatrix}
    \begin{bmatrix}
        x(k) \\
        x_I(k)
    \end{bmatrix}
    +
    \begin{bmatrix}
        B \\
        0
    \end{bmatrix}
    u(k)
    -
    \begin{bmatrix}
        0 \\
        1
    \end{bmatrix}
    r(k)
\end{align*}
\]  

(3)  

\[
\begin{align*}
    u(k) &=
    \begin{bmatrix}
        K & K_I
    \end{bmatrix}
    \begin{bmatrix}
        x(k) \\
        x_I(k)
    \end{bmatrix}
    +
    KNr(k)
\end{align*}
\]  

(4)
1.1 Equivalent Block Diagram for Observer Based Compensator

First by we will assume the following:

\[ \dot{v}_1 + \dot{v}_2 = \dot{x}(k + 1) \]

It follows that:

\[ \dot{v}_1 + \dot{v}_2 = \dot{x}(k + 1) = (A - LC)\dot{x}(k) + L \cdot y(k) + B \cdot u(k) \] (5)

Output of Observer:

\[ u_k(k) = -K\dot{x}(k) \] (6)

By Linearity,

\[ \dot{v}_1 = (A - LC)\dot{x}(k) + B \cdot u(k) \]
\[ \dot{v}_2 = (A - LC)\dot{x}(k) + L \cdot y(k) \] (7)

Converting to Transfer Functions

\[ OB1 = \frac{\dot{V}_1(z)}{U(z)} = K (zI - A + LC)^{-1}B \] (8)

\[ OB2 = \frac{\dot{V}_2(z)}{Y(z)} = K (zI - A + LC)^{-1}L \] (9)

\[ U_k(z) = OB1 + OB2 \] (10)
1.2 Loop Gain Calculation

Through Block Diagram Manipulation we have the following:

Figure 3: Block Diagram of State Observer with Transfer Function Equivalent
Figure 4: Block Diagram of State Observer, Integrator turned into FF and FB term

\[ G(z) \]

Figure 5: Block Diagram of State Observer - Single Loop Structure

\[ \frac{1}{1 + OB1} \cdot G(z) \cdot \left( OB2 + \frac{K_I}{z - 1} \right) \]  \hspace{1cm} (11)

From the figure above, we can see that the loop gain will be:
where,

\[ OB_1 = \frac{\hat{V}_1(z)}{U(z)} = K(zI - A + LC)^{-1}B \]

\[ OB_2 = \frac{\hat{V}_2(z)}{Y(z)} = K(zI - A + LC)^{-1}L \] 

(12)