

# Optimal Control Design of a Voice Coil Head Actuator in a Hard Drive

The challenge: To quickly and accurately position the head of a voice coil actuator in a hard drive

© Maplesoft, a division of Waterloo Maple Inc., 2008

## Executive Summary

## Introduction

## Problem Statement

## 1. Physical Model Representation

## 2. Plant Dynamics Investigation

## 3. Linear Quadratic Gaussian Control Design

## Results



\*\* This application was developed using [Maple](#) and [MapleSim](#)

## ▼ Executive Summary

The ability to quickly access information stored in a hard drive is a crucial feature of modern computers. Since the early 1980s, all hard drives used a voice coil actuator, rather than a stepper motor, to position the read-write head over the surface of a disc track. Voice coil actuators in hard drives have two major advantages over stepper motors: they offer quicker and more accurate positioning and are more tolerant of disc warping and track variations that invariably occur over time.

Accurate positioning of a voice coil actuator head is somewhat complex because of the inability to physically measure the position of the head. As a result, a more complex control strategy had to be implemented to control the movement of the voice coil actuator head. For this application, an optimal state feedback control strategy, linear-quadratic-Gaussian (LQG), was chosen because it can be used to control systems where not all the states are measurable.

The symbolic equations governing the dynamics of the voice coil actuator model (or plant model) were determined by developing a physical model representation of the system in MapleSim. The plant model parameters, which accurately described the behavior of the real-world system, were determined by applying established linearization and system identification techniques.

The LQG controller was designed using the parametric plant model equations. The gain values for the control module, which consisted of a linear-quadratic-regulator (LQR) and a Kalman filter, were determined using user-defined Maple routines. The robustness of the controller was determined by conducting sensitivity analysis and Monte-Carlo simulations on the plant model parameters. The closed-loop system was found to be unaffected by parameter fluctuations.

The step response of the closed-loop system was determined using MapleSim. The symbolic equations governing the controller were incorporated into the physical model representation of the plant model through MapleSim's Custom Component Template. The simulation results showed that many of the oscillations that were present in the operating bandwidth of the open-loop system had been removed. As a result, the closed-loop system offered a speedup gain of 70% over the open-loop response.

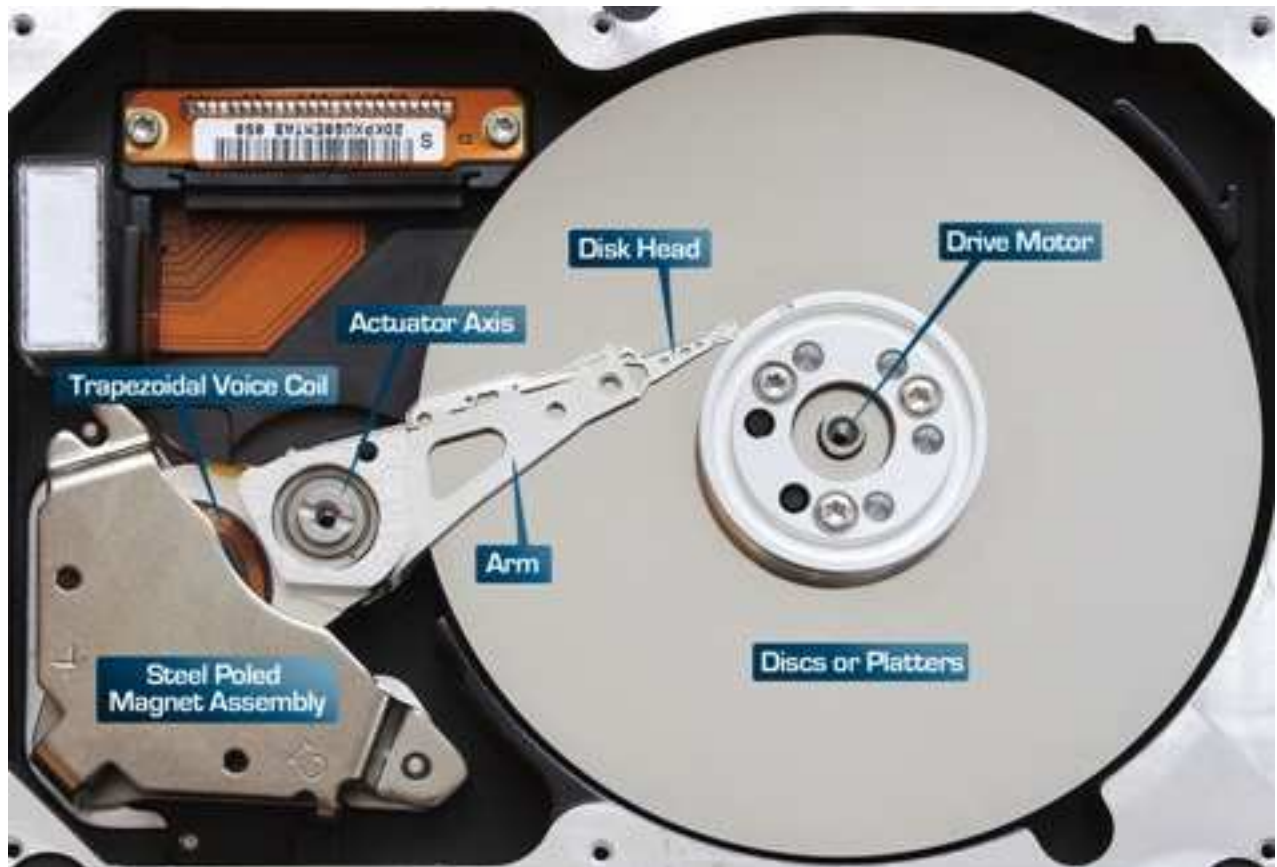
## ▼ Introduction

The use of a stepper motor for positioning in a hard drive was discontinued with the advent of high density hard drives. With densities as great as 30,000 tracks per inch, stepper motors were found to be sluggish and to lack the stability to maneuver accurately between tightly spaced tracks.

The main components of a hard drive as seen in the following figure are: 1) Steel Poled Magnet Assembly, 2) Trapezoidal Voice Coil, 3) Actuator Axis, 4) Arm, 5) Disk Head, 6) Drive Motor, and 7) Discs or Platters.

A voice coil actuator controls the movement of the read-write head following the principle of electromagnetic attraction and repulsion. The coil wrapped around the trapezoidal voice coil is attracted to the magnets within the steel poled magnet assembly. When a current is fed to the coil, an electromagnetic field is generated. This causes the arm to move in a particular direction based on the relative attractive and repulsive forces of the permanent magnet. By controlling the flow of current to the coil, both magnitude and direction, the position of the arm can be adjusted very finely.

This type of fine positioning is not possible in a stepper motor because the stepper size of the pulse signal is fixed.



## ▼ Problem Statement

This document examines the steps involved in quickly and accurately controlling the position of the voice coil actuator head. The design of the closed-loop system, including the approach used to investigate the dynamics of the plant model and design the mathematical model of the controller, will be explained in detail in the following sections.

## ▼ 1. Physical Model Representation

The MapleSim model shown in Figure 1 is a physical model representation of a controlled voice coil head actuator system in a hard drive. As shown in the diagram, the physical model consists of a controller model and a plant model. The equations governing the LQG controller were determined by analyzing the dynamics of the plant model system. As can be seen in the figure below, the control equations were incorporated into the physical model as a MapleSim custom component block. The custom component controller block and the plant model together formed a closed-loop system.

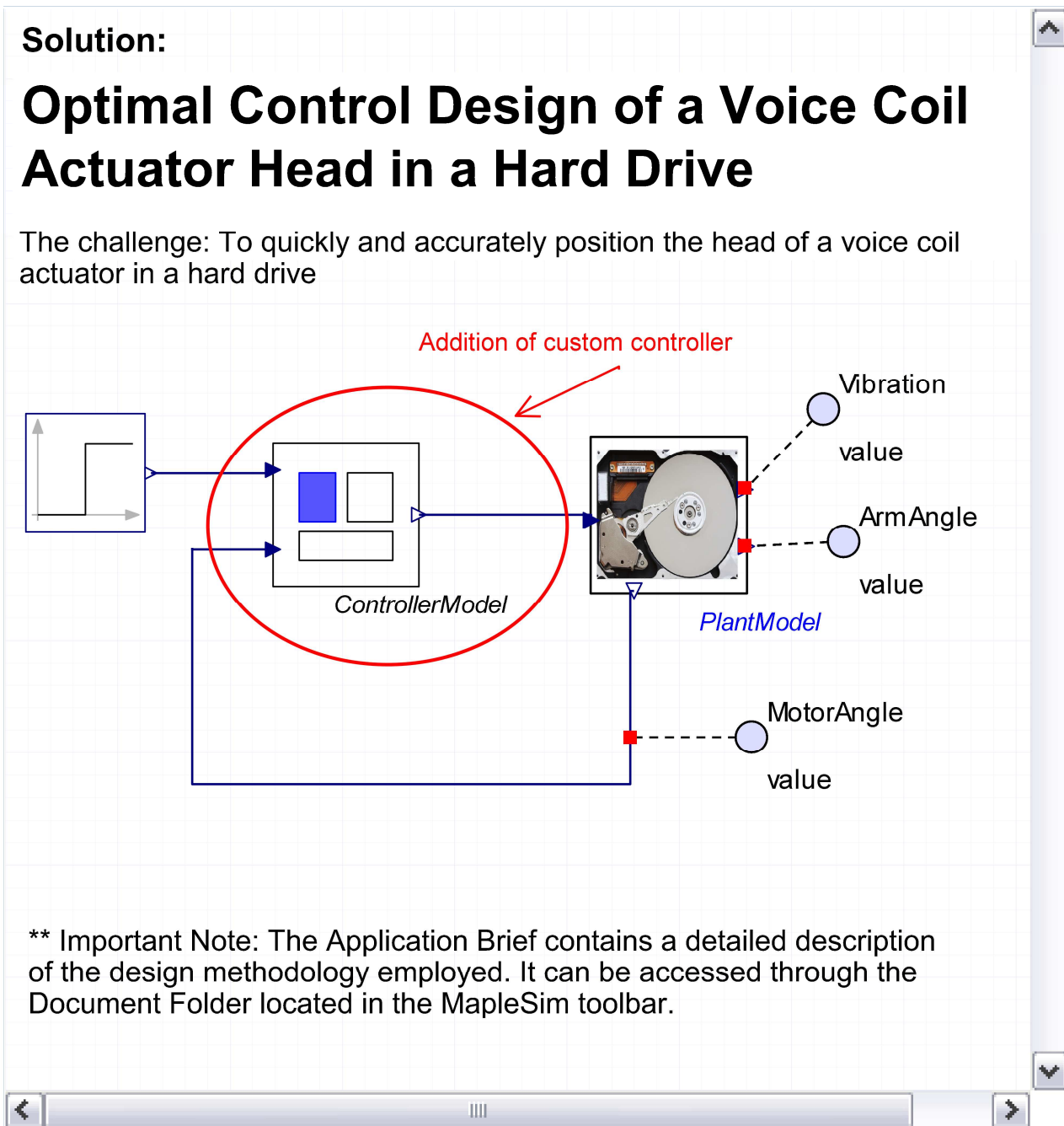


Figure 1: Physical model representation of a Controlled Voice Coil Head Actuator

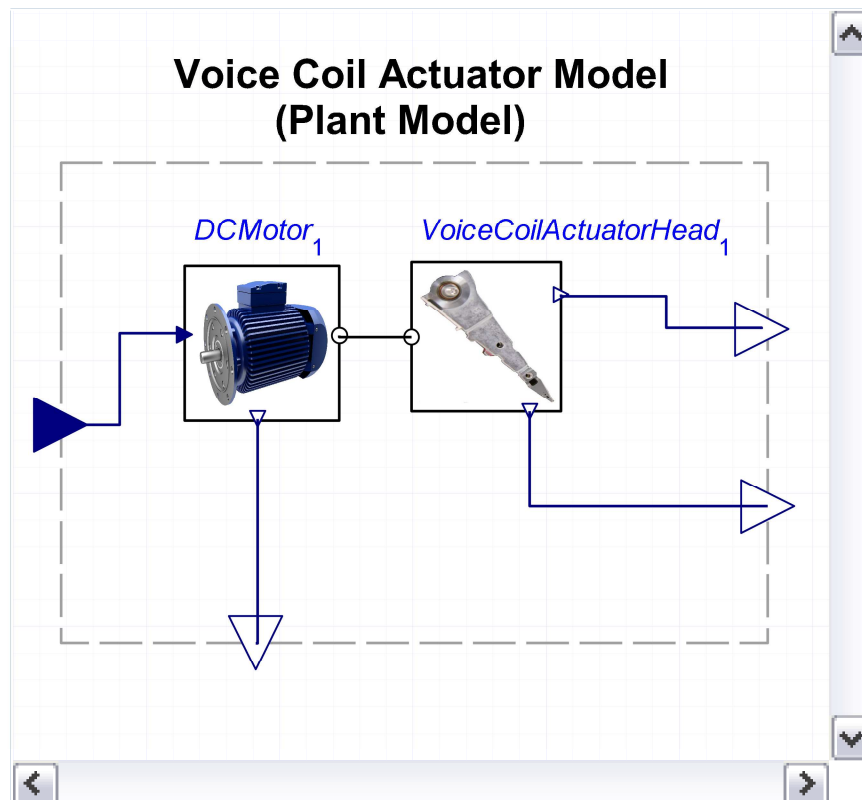
## ▼ 2. Plant Dynamics Investigation

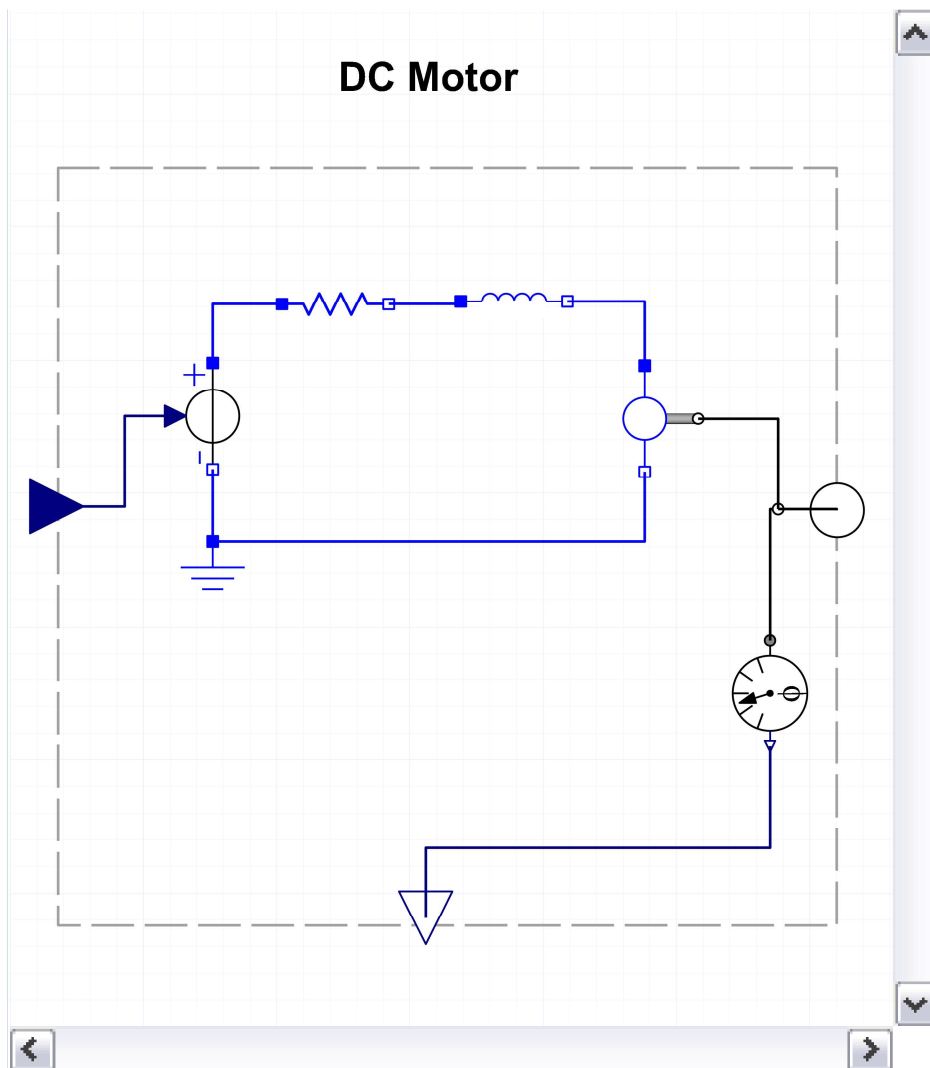
The three diagrams below illustrate how the plant model was designed in MapleSim. The plant model consists of two submodels: 1) the DC Motor submodel, and 2) the Voice Coil Actuator Head submodel.

- The DC Motor submodel was modeled using a simplified model of a DC motor as an equivalent electric circuit.
- The Voice Coil Actuator Head submodel was built by cascading a revolute joint and a flexible beam component. These pre-built components are included in MapleSim's Multibody library.

It is important to note that in the real world the movement of the head across the discs or platters of the hard drive is driven by a steel poled magnet assembly. Since a steel poled magnet assembly has similar dynamic behavior as that of a DC Motor, within a specific operating range, it is common practice to model the electromagnetic drive component of a system by a DC Motor. The main advantage of doing so are to reduce the complexity of the model and to decrease the simulation time.

The choice of parameter values for the inductor and EMF components in the DC Motor submodel, which accurately model the dynamics of the steel poled magnet assembly, will be described in the next section.





# Voice Coil Actuator Head Model

### Extract Plant Model Equations

$$\left[ \frac{d}{dt} \text{'DFPSysInst.theta\_flange\_b\_dot'}(t) - \left( \frac{d}{dt} \text{'DFPSysInst.theta\_VoiceCoilActuatorHead1\_R3\_dot'}(t) \right) = 0, \right. \\ \left( \frac{1}{1000000000000000000} \cos(\text{'DFPSysInst.theta\_VoiceCoilActuatorHead1\_R3\_'}(t)) \right. \\ \sin(\text{'DFPSysInst.theta\_VoiceCoilActuatorHead1\_R3\_'}(t)) \text{'DFPSysInst.vf\_VoiceCoilActuatorHead1\_FB2\_1'}(t) \\ \left. + \frac{243}{25000000000} \sin(\text{'DFPSysInst.theta\_VoiceCoilActuatorHead1\_R3\_'}(t))^2 \right. \\ \left. + \frac{27}{12500000} \right]$$

### Nonlinear equations governing the dynamics of the plant model

## ▼ Equation Linearization

These equations were linearized using user-defined algorithms developed in Maple. The equations were linearized to employ standard linear control design methods. The equations were linearized by setting the equilibrium point of the state variables and input variables to 0.

$$\text{stateeqpoint} := [I2.i(t) = 0, \text{DFPSubsys1inst.theta\_R3}(t) = 0, \\ \text{DFPSubsys1inst.vf\_FB2\_1}(t) = 0, \text{DFPSubsys1inst.theta\_R3\_dot}(t) = 0, \\ \text{DFPSubsys1inst.vf\_FB2\_1\_dot}(t) = 0]:$$

$$\text{inpupteqpoint} := [\text{RealInput4}(t) = 0]:$$

The results of the linearization *LinearizedSystem\_DE*, were left in symbolic form to facilitate system identification and parameter sensitivity analysis (see the [System Identification](#) and [Open-Loop Parameter Sensitivity Analysis](#) sections).

$$\begin{aligned} \text{LinearizedSystem\_DE} := & \left[ \frac{d}{dt} I2.i(t) = -\frac{R4.R I2.i(t)}{I2.L} \right. \\ & - \frac{EMF2.k \text{DFPSubsys1inst.theta\_R3\_dot}(t)}{I2.L} + \frac{\text{RealInput4}(t)}{I2.L}, \\ & \frac{d}{dt} \text{DFPSubsys1inst.theta\_R3}(t) = \text{DFPSubsys1inst.theta\_R3\_dot}(t), \\ & \frac{d}{dt} \text{DFPSubsys1inst.vf\_FB2\_1}(t) = \text{DFPSubsys1inst.vf\_FB2\_1\_dot}(t), \\ & \frac{d}{dt} \text{DFPSubsys1inst.theta\_R3\_dot}(t) = \frac{20054000000000}{181404729} EMF2.k I2.i(t) \\ & + \frac{80360000000000}{60468243} \text{DFPSubsys1inst.vf\_FB2\_1}(t), \frac{d}{dt} \text{DFPSubsys1inst.vf\_FB2\_1\_dot}(t) \\ & = -\frac{251125000000000}{181404729} EMF2.k I2.i(t) \\ & - \frac{4024000000000000}{181404729} \text{DFPSubsys1inst.vf\_FB2\_1}(t), RO3(t) \\ & = \frac{3}{25} \text{DFPSubsys1inst.vf\_FB2\_1}(t), RO5(t) = 'AS3::CB'.k \text{DFPSubsys1inst.theta\_R3}(t), \\ & \left. RO6(t) = \text{DFPSubsys1inst.theta\_R3}(t) + \frac{3}{25} \text{DFPSubsys1inst.vf\_FB2\_1}(t) \right]: \end{aligned}$$

The differential equations defined in *LinearizedSystem\_DE* were converted into symbolic state-space form.

$$\begin{aligned} \text{LinearizedSystem\_SS} := & \text{DynamicSystems}[\text{StateSpace}](\text{LinearizedSystem\_DE}, \text{outputvariable} \\ & = [RO3(t), RO5(t), RO6(t)], \text{inputvariable} = [\text{RealInput4}(t)], \text{statevariable} = [I2.i(t), \\ & \text{DFPSubsys1inst.theta\_R3}(t), \text{DFPSubsys1inst.vf\_FB2\_1}(t), \\ & \text{DFPSubsys1inst.theta\_R3\_dot}(t), \text{DFPSubsys1inst.vf\_FB2\_1\_dot}(t)]) \end{aligned}$$



## State Space

continuous

3 output(s); 1 input(s); 5 state(s)

inputvariable = [ *RealInput4*(*t*) ]

outputvariable = [ *RO3*(*t*), *RO5*(*t*), *RO6*(*t*) ]

statevariable = [ *I2.i*(*t*), *DFPSysinst.theta\_R3*(*t*), *DFPSysinst.vf\_FB2\_1*(*t*),  
*DFPSysinst.theta\_R3\_dot*(*t*), *DFPSysinst.vf\_FB2\_1\_dot*(*t*) ]

(1)

## ▼ System Identification

As mentioned previously, in the real world the force driving the voice coil head is derived from a steel poled magnet assembly. As a result, the parameter values of the DC motor -- namely the inductor and the EMF component, which yield similar dynamics to that of the magnet assembly -- need to be identified. The parameter value for the resistor is known.

The impulse response of the real-world system to an impulse at  $t=0.01$ s is shown in the plots below. The data was collected using a standard data acquisition card.

*ExperimentalData* := *MapleSim*[*Attachments*][*Get*]("Simulation0", "ExperimentalData.csv")

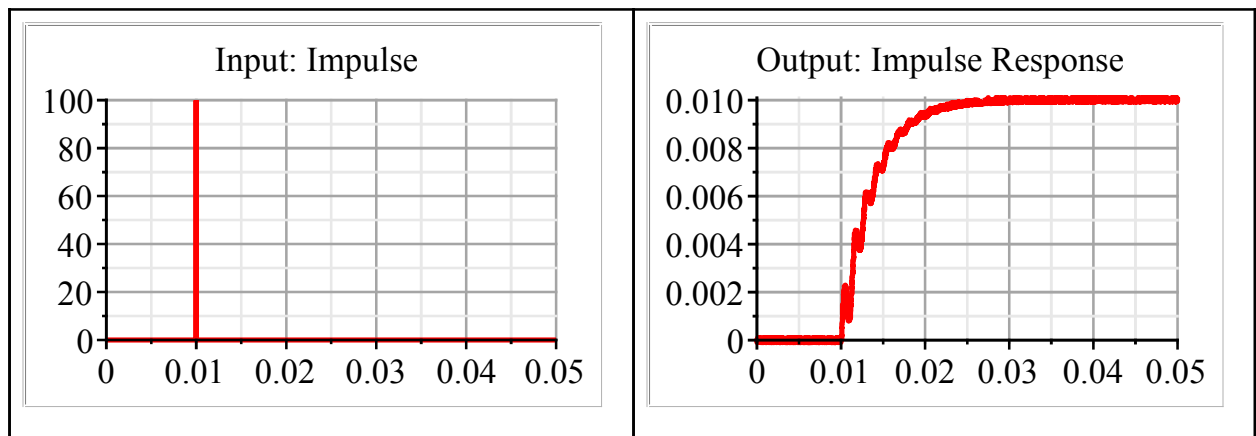
5001 x 3 Matrix  
 Data Type: anything  
 Storage: rectangular  
 Order: Fortran\_order

(2)

*Time* := *ExperimentalData*[1 .. -1, 1] :

*Input* := *ExperimentalData*[1 .. -1, 2] :

*Output* := *ExperimentalData*[1 .. -1, 3] :



Using user-defined algorithms, the experimental data was manipulated into a form, which exposed the input-output dynamics of the magnet assembly:





## ▼ EMF parameter sensitivity

The Bode sensitivity function with respect to the EMF parameter is calculated as follows:

$$\left( \frac{\partial}{\partial 'EMF2.k'} SensitivityTF \right) \cdot \frac{'EMF2.k'}{SensitivityTF} \quad (7)$$

$$\left( \left( (100000000000000000000 - 90729000000000 s^2) / (1632642561 I2.L s^5 \right. \right.$$

$$\left. \left. + 1632642561 R4.R s^4 + (1804860000000000 EMF2.k^2 \right. \right.$$

$$\left. \left. + 36216000000000000 I2.L) s^3 + 36216000000000000 R4.R s^2 \right. \right.$$

$$\left. \left. + 100000000000000000000 EMF2.k^2 s \right) - ( (-90729000000000 EMF2.k s^2 \right.$$

$$\left. \left. + 100000000000000000000 EMF2.k) (360972000000000 EMF2.k s^3 \right. \right.$$

$$\left. \left. + 200000000000000000000 EMF2.k s) \right) / (1632642561 I2.L s^5 \right.$$

$$\left. \left. + 1632642561 R4.R s^4 + (1804860000000000 EMF2.k^2 \right. \right.$$

$$\left. \left. + 362160000000000000 I2.L) s^3 + 36216000000000000 R4.R s^2 \right. \right.$$

$$\left. \left. + 100000000000000000000 EMF2.k^2 s \right)^2 \right) EMF2.k (1632642561 I2.L s^5 \right.$$

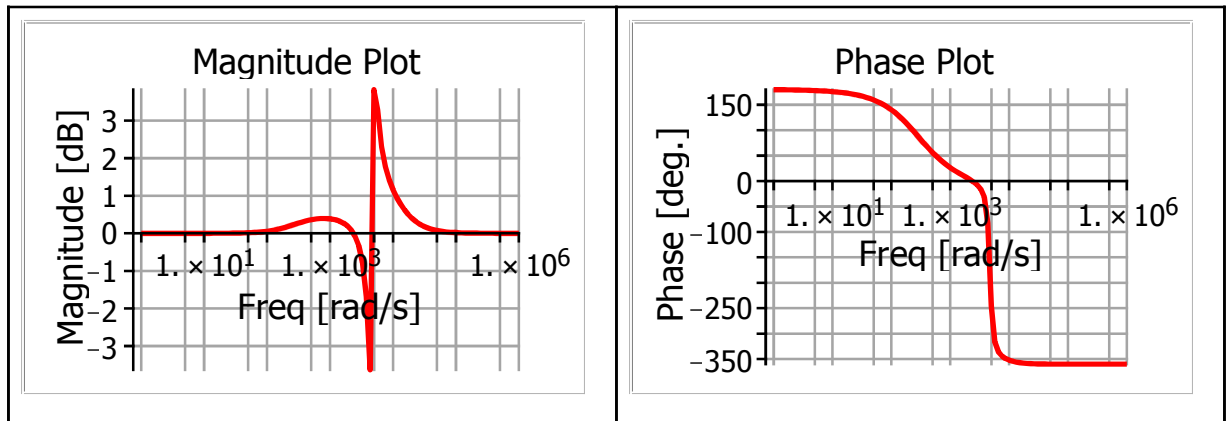
$$\left. \left. + 1632642561 R4.R s^4 + (1804860000000000 EMF2.k^2 \right. \right.$$

$$\left. \left. + 362160000000000000 I2.L) s^3 + 36216000000000000 R4.R s^2 \right. \right.$$

$$\left. \left. + 100000000000000000000 EMF2.k^2 s \right) \right) / ( (-90729000000000 EMF2.k s^2 \right.$$

$$\left. \left. + 100000000000000000000 EMF2.k) \right)$$

$S_{EMFK} := DynamicSystems[TransferFunction]((7)) :$

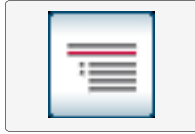


As shown in the plot, the sensitivity of the system with respect to the EMF parameter value is large for all frequencies.

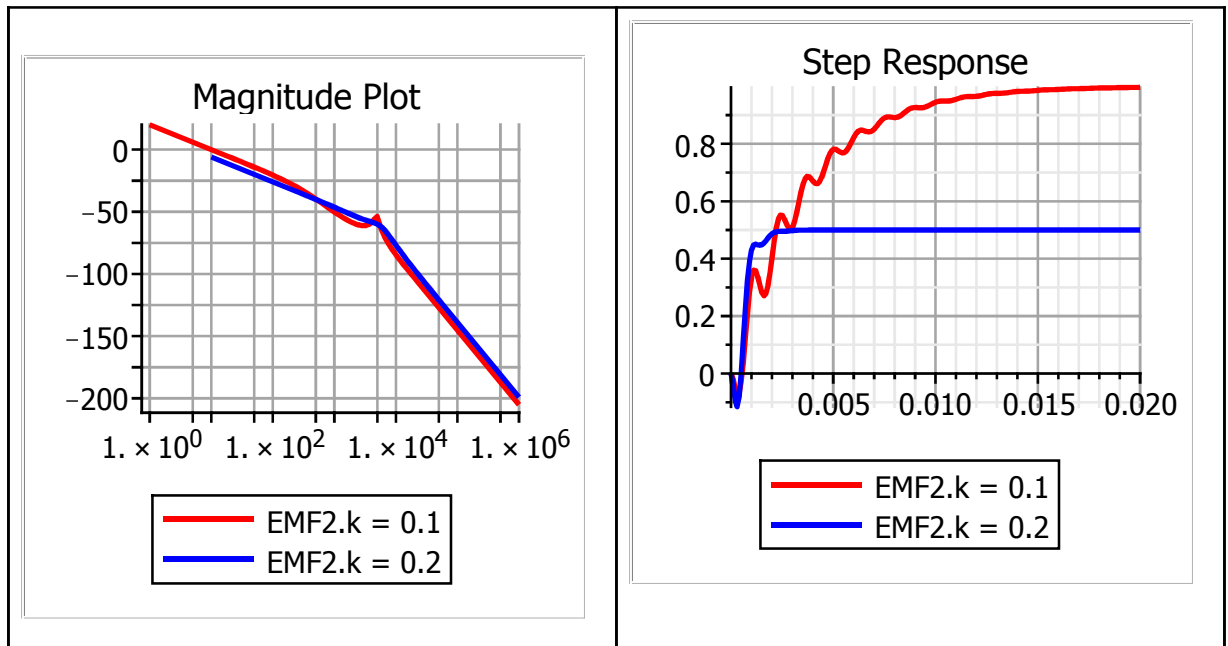
The effects of changing the EMF parameter value from 0.1 to 0.2 can be seen in the magnitude plot and step response plots below.

```
NewDCMotorParameters := [ `EMF2.k` = 0.2, `I2.L` = 0.0001, `R4.R` = 1, `AS3::CB'.k` = 1.0 ]:
```

*Note: The commands used to define the plots can be found by expanding the code edit region*



#Magnitude and Step Response



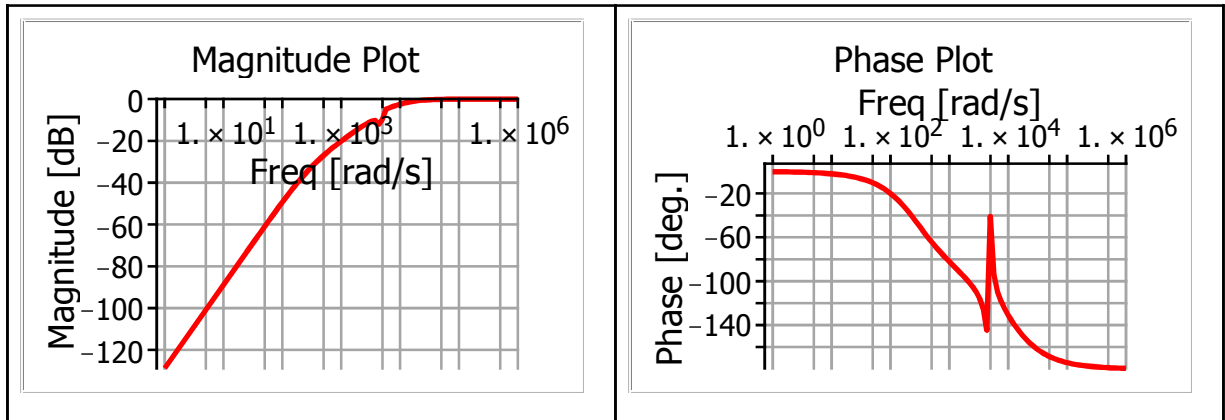
As expected, an increase in the EMF parameter resulted in a large change in the system response.

## ▼ Inductor parameter sensitivity

The Bode sensitivity function with respect to the inductor parameter is calculated as follows:

$$\left( \frac{\partial}{\partial I2.L} \text{SensitivityTF} \right) \cdot \frac{I2.L}{\text{SensitivityTF}} - \left( (1632642561 s^5 + 36216000000000000 s^3) I2.L \right) / (1632642561 I2.L s^5 + 1632642561 R4.R s^4 + (1804860000000000 EMF2.k^2 + 36216000000000000 I2.L) s^3 + 36216000000000000 R4.R s^2 + 100000000000000000000 EMF2.k^2 s) \quad (8)$$

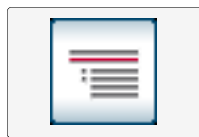
```
SI2.L := DynamicSystems[TransferFunction]((8)) :
```



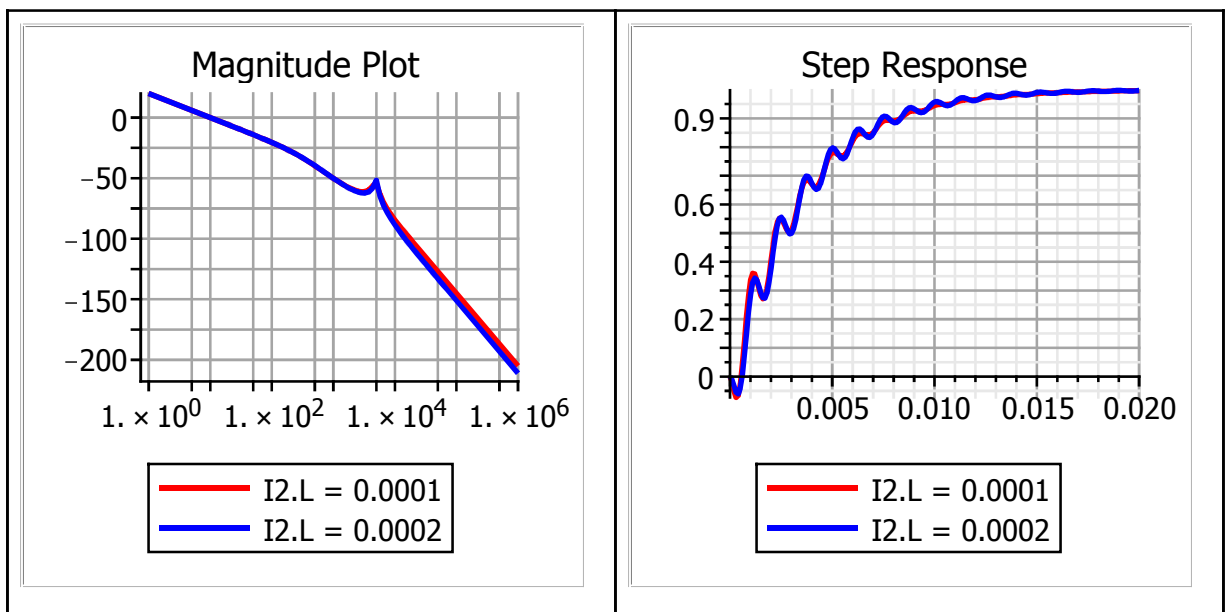
As shown in the plot, the sensitivity of the system with respect to the inductor parameter value is large at high frequencies.

The effects of changing the inductor parameter value from 0.0001 to 0.0002 can be seen in the magnitude plot and step response plots below.

*NewDCMotorParameters1* := [ 'EMF2.k'=0.1, 'I2.L'=0.0002, 'R4.R'=1, 'AS3::CB'.k'=1.0 ] :



#Magnitude and Step Response



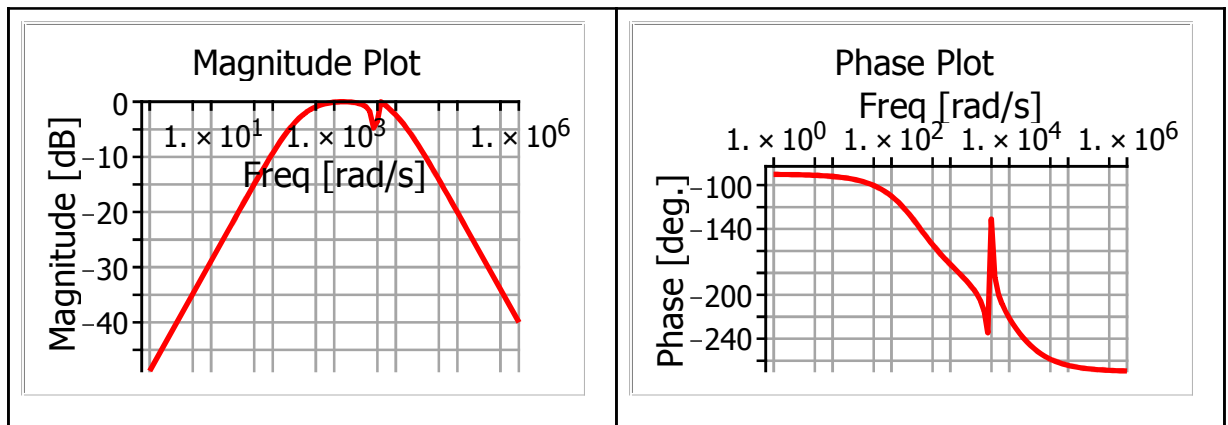
Based on the plot above, we can see that variations in the inductance parameter are only affected at high frequencies.

## ▼ Resistor parameter sensitivity

The Bode sensitivity function with respect to the resistor parameter is calculated as follows:

$$\left( \frac{\partial}{\partial 'R4.R'} \text{SensitivityTF} \right) \cdot \frac{'R4.R'}{\text{SensitivityTF}} - \left( (1632642561 s^4 + 36216000000000000 s^2) R4.R \right) / (1632642561 I2.L s^5 + 1632642561 R4.R s^4 + (1804860000000000 EMF2.k^2 + 36216000000000000 I2.L) s^3 + 36216000000000000 R4.R s^2 + 10000000000000000000 EMF2.k^2 s) \quad (9)$$

$S_{R4.R} := \text{DynamicSystems}[\text{TransferFunction}](9) :$



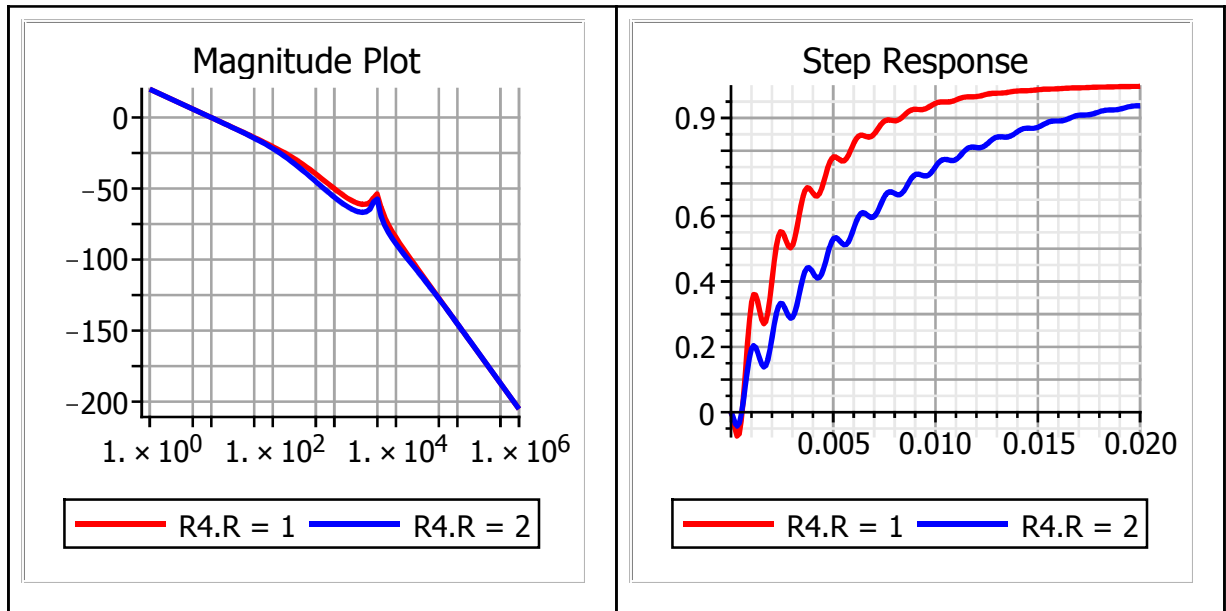
As shown in the plot, the sensitivity of the system with respect to the resistor parameter value is large at the operating bandwidth.

The effects of changing the resistance parameter value from 1 to 2 are shown in the magnitude plot and step response plots below.

$\text{NewDCMotorParameters2} := [ 'EMF2.k' = 0.1, 'I2.L' = 0.0001, 'R4.R' = 2, 'AS3::CB'.k' = 1.0 ] :$



## #Magnitude and Step Response



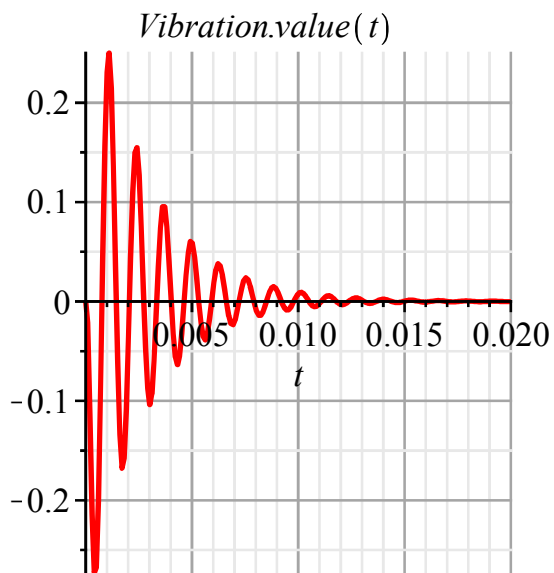
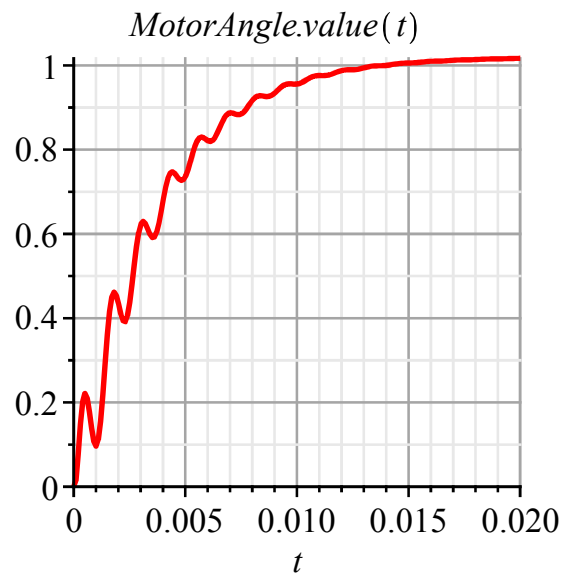
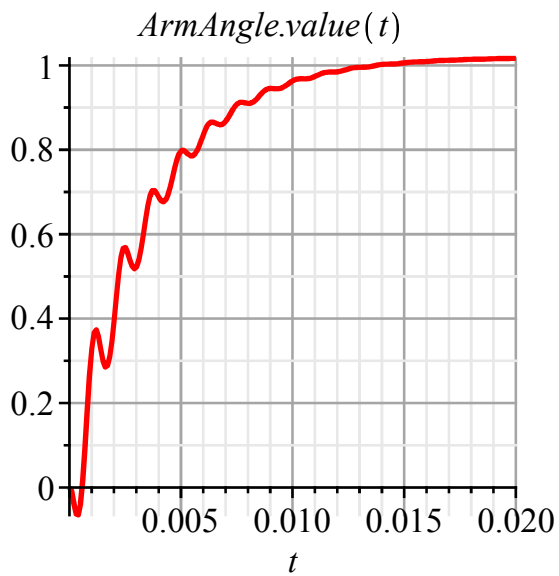
Based on the plot above, we can see that variations in the resistor parameter lead to slower response times.

## ▼ Open-Loop Response

The system response of the plant model (in isolation from the controller) using *DCMotorParameters* are shown in the plots below. The oscillations that are visible in the plots make accurate positioning very difficult to achieve. To achieve quicker and more reliable positioning, a controller is needed to remove some of the oscillations from the operating bandwidth of the system.



## Plant Simulation Results



### ▼ 3. Linear Quadratic Gaussian (LQG) Control Design

#### ▼ LQG Controller Design

An LQG controller is created by coupling an observer block, such as a Kalman State Estimator (Kalman Filter), with a Linear-Quadratic-Regulator (LQR). An observer block was required to estimate the unmeasured states of the plant model. In this application, the unmeasured state refers to the angle position of the arm because it cannot be physically measured (only the angle position of the motor drive can be measured). If it were measurable, we could eliminate the observer block and replace the LQG control strategy in favor of a more simple controller such as an LQR controller.

An LQR controller is a standard full-state feedback controller, which requires all of the states of the plant model to be known. The Kalman filter is an efficient continuous filter, which estimates the current value of the plant states based on the estimated state from the previous time step, the current measurement, and the input signal. Once the values of the unmeasured states are estimated, they can be fed into the LQR controller.

The block diagram representation of the model-based control strategy is depicted in Figure 2. The design of the LQR controller, which consists of a state feedback gain block and a pre-filtering gain block, and the Kalman filter will be described in the upcoming sections. The separation principle guarantees that the controller and observer blocks can be designed and computed independently of each other. It is important to note that many of the algorithms that were developed to calculate the controller and observer gains were created by using user-defined software routines developed in Maple.

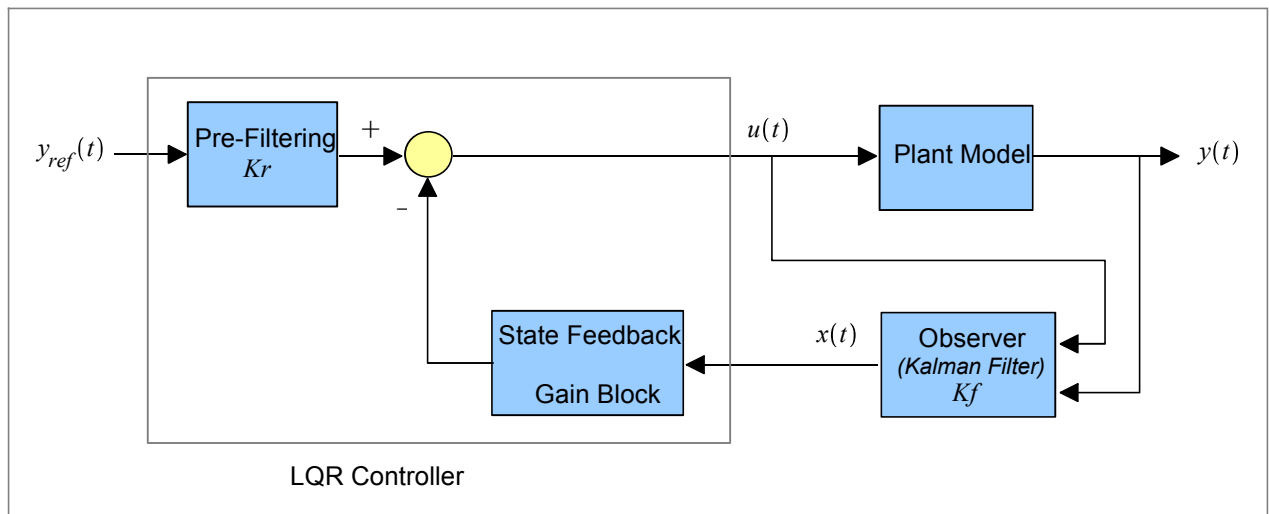


Figure 2: LQG design strategy block diagram

#### ▼ Step 1: LQR Controller Design

The design of the LQR controller is based on the equations governing the plant model. The linearized plant model equations as derived in *LinearizedSystem\_SS* are:

```
PrintSystem(LinearizedSystem_SS)
```

## State Space

continuous

3 output(s); 1 input(s); 5 state(s)

inputvariable = [*RealInput4*(*t*) ]

outputvariable = [*RO3*(*t*), *RO5*(*t*), *RO6*(*t*) ]

statevariable = [*I2.i*(*t*), *DFPSubsys1inst.theta\_R3*(*t*),  
*DFPSubsys1inst.vf\_FB2\_1*(*t*), *DFPSubsys1inst.theta\_R3\_dot*(*t*),  
*DFPSubsys1inst.vf\_FB2\_1\_dot*(*t*) ]

$$\begin{aligned}
 \mathbf{a} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{80360000000000}{60468243} & 0 & \frac{20054000000000 \text{ EMF2.k}}{181404729} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{4024000000000000}{181404729} & 0 & -\frac{25112500000000 \text{ EMF2.k}}{181404729} \\ 0 & -\frac{\text{EMF2.k}}{I2.L} & 0 & 0 & -\frac{R4.R}{I2.L} \end{bmatrix} \\
 \mathbf{b} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I2.L} \end{bmatrix} \\
 \mathbf{c} &= \begin{bmatrix} 0 & 0 & \frac{3}{25} & 0 & 0 \\ 'AS3::CB'.k & 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{3}{25} & 0 & 0 \end{bmatrix} \\
 \mathbf{d} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{10}$$

As mentioned previously, the first step in implementing a LQG control strategy is to design an LQR controller. The LQR algorithm is, at its core, just an automated way of finding an appropriate state-feedback controller.

Using user-defined algorithms, the gain values for the pre-filtering block,  $K_r$ , and the state feedback gain block,  $K_c$ , were found to be the following:

$K_r := [688.932330743120132] :$

$K_c := [688.93233074312006573859860437762214622832242627265,$   
 $0.80550499997135537655787848078577937523862968172366,$   
 $22.160712974762312223886404357194444118089378945251,$   
 $0.044416647205194564666682746386651570386716496548781,$   
 $0.2454696917613271877958156823410048270007793496841] :$

The state space representation for the closed-loop system with the LQR controller was found to be:

$LQRClosedLoopDefinition := DynamicSystems[StateSpace](a\_lqr, b\_lqr, c\_lqr, d\_lqr,$   
 $outputvariable = [RO3(t), RO5(t), RO6(t)], inputvariable = [RealInput4(t)],$   
 $statevariable = [I2.i(t), DFPSubsys1inst.theta\_R3(t), DFPSubsys1inst.vf\_FB2\_1(t),$   
 $DFPSubsys1inst.theta\_R3\_dot(t), DFPSubsys1inst.vf\_FB2\_1\_dot(t)])$

### State Space

continuous

3 output(s); 1 input(s); 5 state(s)

inputvariable =  $[RealInput4(t)]$

(11)

outputvariable =  $[RO3(t), RO5(t), RO6(t)]$

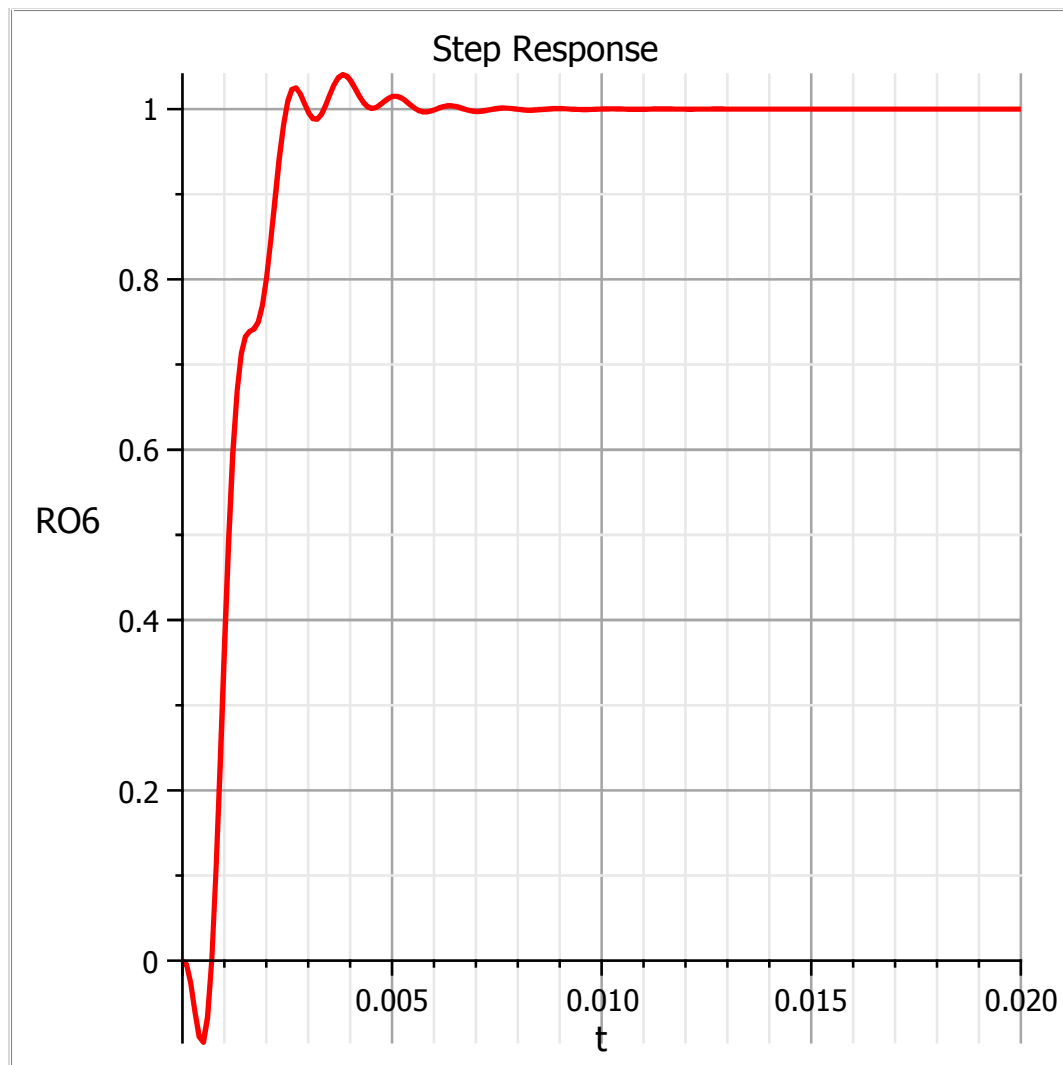
statevariable =  $[I2.i(t), DFPSubsys1inst.theta\_R3(t),$   
 $DFPSubsys1inst.vf\_FB2\_1(t), DFPSubsys1inst.theta\_R3\_dot(t),$   
 $DFPSubsys1inst.vf\_FB2\_1\_dot(t)]$

The state space matrices (A, B, C and D) used to define the LQR controller can be found by expanding the following code edit region.



# State Space Definition

The step response of the closed-loop system (plant + LQR controller) is plotted below, using the parameters defined earlier in *DCMotorParameters*.



As shown in the step response plot, the controller removed many of the oscillations that were present in the operating bandwidth of the plant model.

## ▼ Step 2: Kalman State Estimator Design

In step 1, the gain values for the LQR controller blocks were defined. However, this definition assumed that all of the states for the plant model were available. This is not the case for this application because the angle position of the arm is not measurable. To compensate for this, a Kalman filter block was incorporated into the design of the controller module to estimate the value of the unmeasurable states.

A Kalman filter is formulated by assuming that noise (both process and measurement noise) on the output signal are uncorrelated. The process noise describes the mismatch between the model and the physical system, while the measurement noise is an estimate of the noise at the output.

The covariance matrices for the process and measurement noise are defined below.

### Process Noise

The covariance matrix defined in  $V$  is based on a classic formula generally used as a first approach.

The use of an identity matrix for  $V0$  indicates that the process noise on each of the states is roughly of the same relative weight.

$$V0 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} :$$

$$qI := 1 :$$

$$V1 := \text{subs}(DCMotorParameters, V0 + qI^2 \cdot \text{LinearizedSystem\_SS} \cdot b \cdot \text{LinearizedSystem\_SS} \cdot b^{\%T})$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1.00000001 \cdot 10^8 \end{bmatrix} \quad (12)$$

#### Measurement Noise

In this application, there is only one measured output. As a result, a 1x1 covariance matrix was defined in  $W1$  to represent the level of noise on the output.

$$W1 := \begin{bmatrix} 0.00000001 \end{bmatrix} :$$

Using proprietary software, we could use the values defined by the covariance matrices in  $V1$  and  $W1$  to calculate the Kalman filter gain ( $Kf$ ).

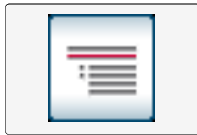
$$Kf := \begin{bmatrix} 15936.0644386437434 \\ 7.69790748963028788 \cdot 10^7 \\ -71589.1680138910888 \\ -9.33536895842888236 \cdot 10^8 \\ 8.32664854267932102 \cdot 10^6 \end{bmatrix} :$$

With the Kalman filter gain defined, the equation representing the LQG controller (LQR controller + Observer) can be obtained. The state space representation of the LQG controller is defined in the variable  $LQGcontroller$ :

$$LQGcontroller := \text{DynamicSystems}[StateSpace](a\_lqg, b\_lqg, c\_lqg, d\_lqg)$$

$$\begin{aligned}
 &\text{State Space} \\
 &\text{continuous} \\
 &1 \text{ output(s); } 2 \text{ input(s); } 5 \text{ state(s)} \\
 &\text{inputvariable} = [u1(t), u2(t)] \\
 &\text{outputvariable} = [y1(t)] \\
 &\text{statevariable} = [x1(t), x2(t), x3(t), x4(t), x5(t)]
 \end{aligned}
 \tag{13}$$

The state space matrices (A, B, C, and D) used to define the LQG controller can be found by expanding the following code edit region.



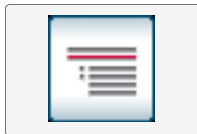
```
# State Space Definition
```

The state space representation of the closed-loop system, which includes the LQG controller and the plant model, is defined in the variable *LQGClosedLoopDefinition*:

```
LQGClosedLoopDefinition := DynamicSystems[StateSpace](a_cl, b_cl, c_cl, d_cl,
  outputvariable = [y1(t), y2(t), y3(t)], inputvariable = [u1(t)], statevariable = [x1(t),
  x2(t), x3(t), x4(t), x5(t), x6(t), x7(t), x8(t), x9(t), x10(t)])
```

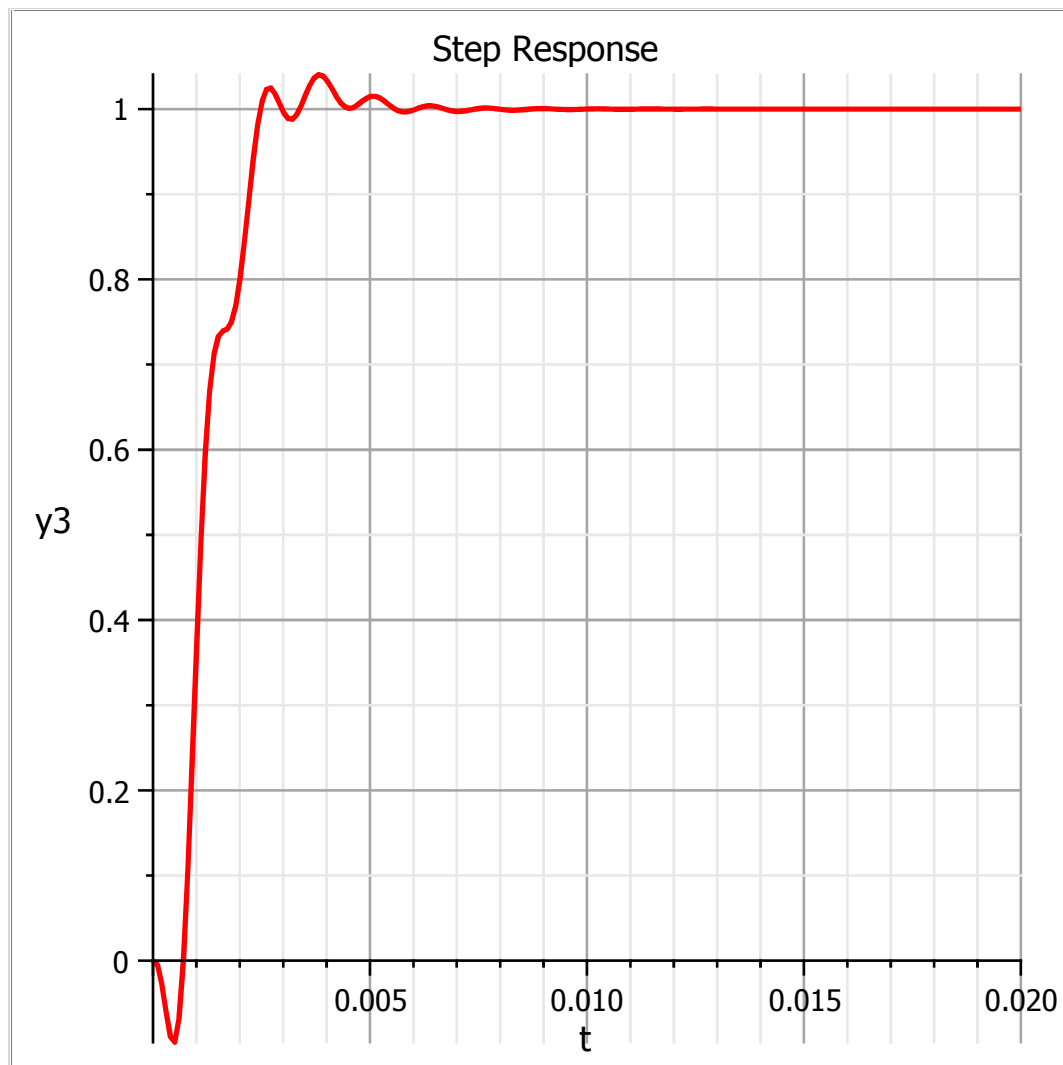
$$\begin{aligned}
 &\text{State Space} \\
 &\text{continuous} \\
 &3 \text{ output(s); } 1 \text{ input(s); } 10 \text{ state(s)} \\
 &\text{inputvariable} = [u1(t)] \\
 &\text{outputvariable} = [y1(t), y2(t), y3(t)] \\
 &\text{statevariable} = [x1(t), x2(t), x3(t), x4(t), x5(t), x6(t), x7(t), x8(t), x9(t), x10(t)]
 \end{aligned}
 \tag{14}$$

The state space matrices (A, B, C, and D) used to define the closed-loop system can be found by expanding the following code edit region.



```
# State Space Definition
```

The step response of the closed-loop system is plotted below:



As expected, the step response with the inclusion of an LQG controller yields similar results to that of the LQR controller alone.

## ▼ Closed-Loop Parameter Sensitivity Analysis

Since the DC motor parameters still appear in the closed-loop system equations, we can investigate the sensitivity of the controlled system with respect to changes in the parameter definition. Analyzing the sensitivity of the closed-loop DC motor parameters provides us with insight into the robustness of the controller with respect to uncertainty in the parameter values.

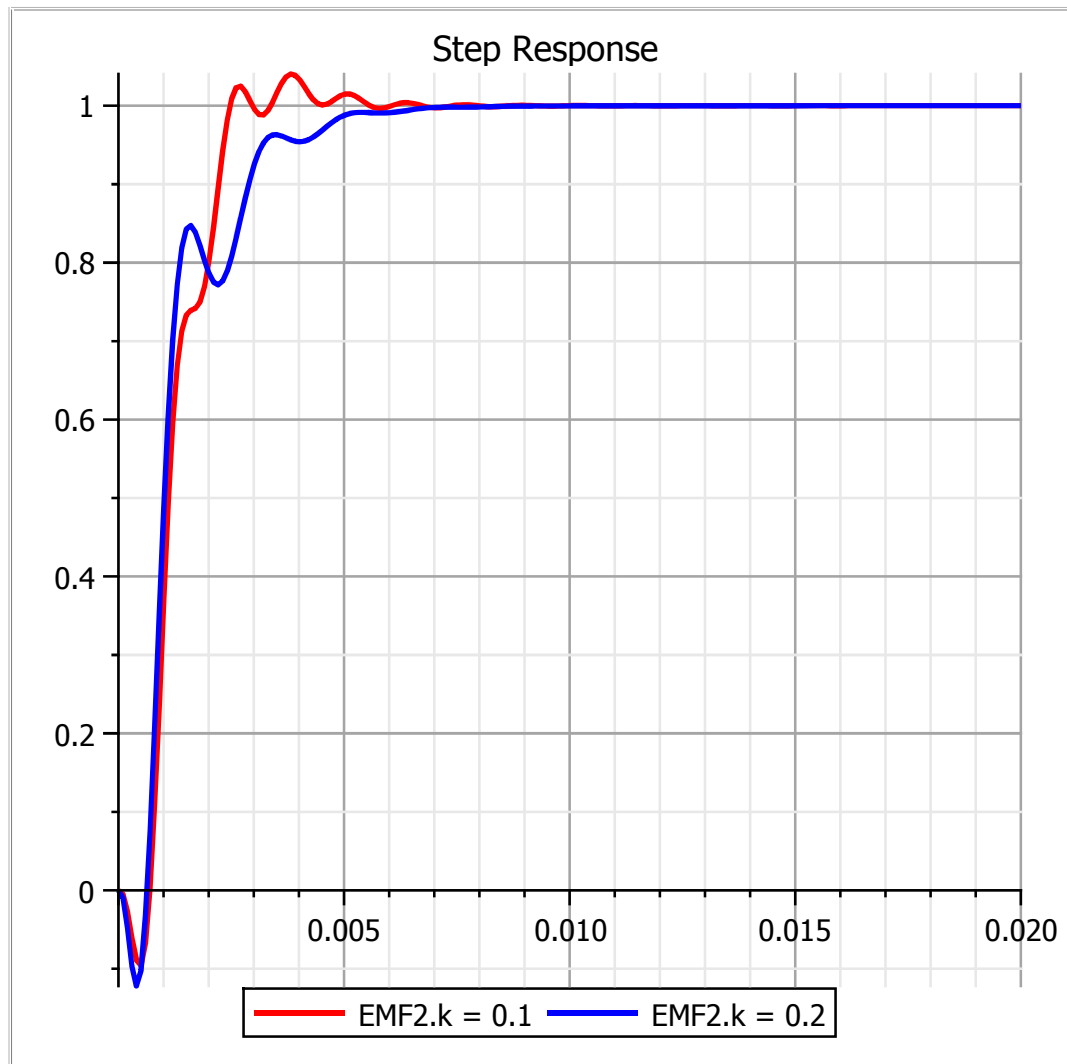
For this section, the same nominal and adjusted DC motor parameter values that were defined in the open-loop sensitivity section (see the [Open-Loop Parameter Sensitivity Analysis](#) section) will be used. In addition, Monte-Carlo simulations for the EMF, inductor, and resistor parameter values will be calculated to ensure that parameter variations and fluctuations that may occur in the real world result in minimal disturbances to the overall closed-loop system response.

As will be shown, the effects of parameter variations on the closed-loop system will be minimal in comparison to that of the open-loop system. This is expected since the system is now controlled. This finding proves that the LQG controller developed in the previous section is robust.



## ▼ EMF parameter sensitivity

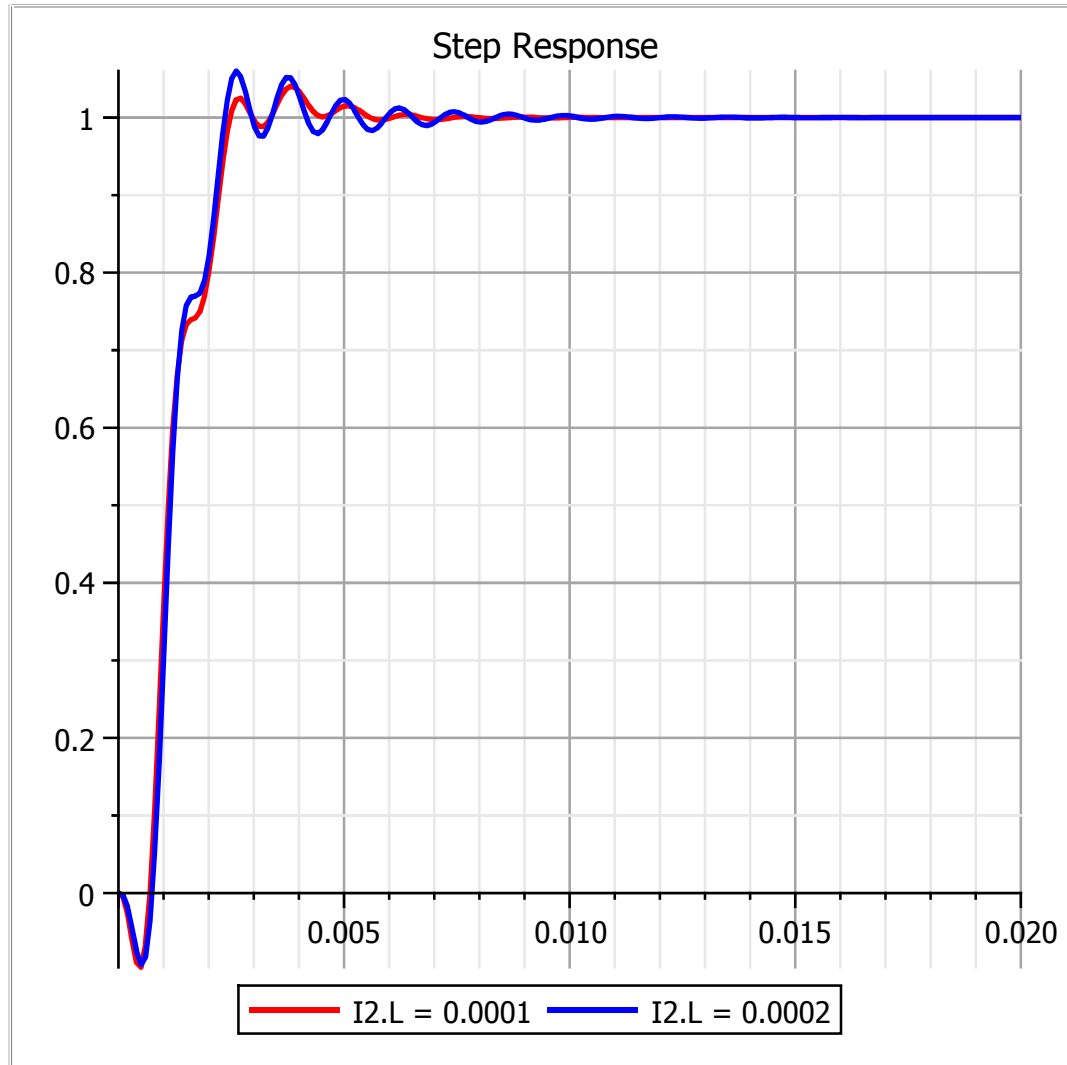
The sensitivity of the system to a change in EMF parameter value is shown in the following step response plot. The EMF parameter value was perturbed from 0.1 Nm/A to 0.2 Nm/A.



As shown, the effects of changing the EMF parameter value in the closed-loop system are very small compared to that of the open-loop system.

## ▼ Inductor parameter sensitivity

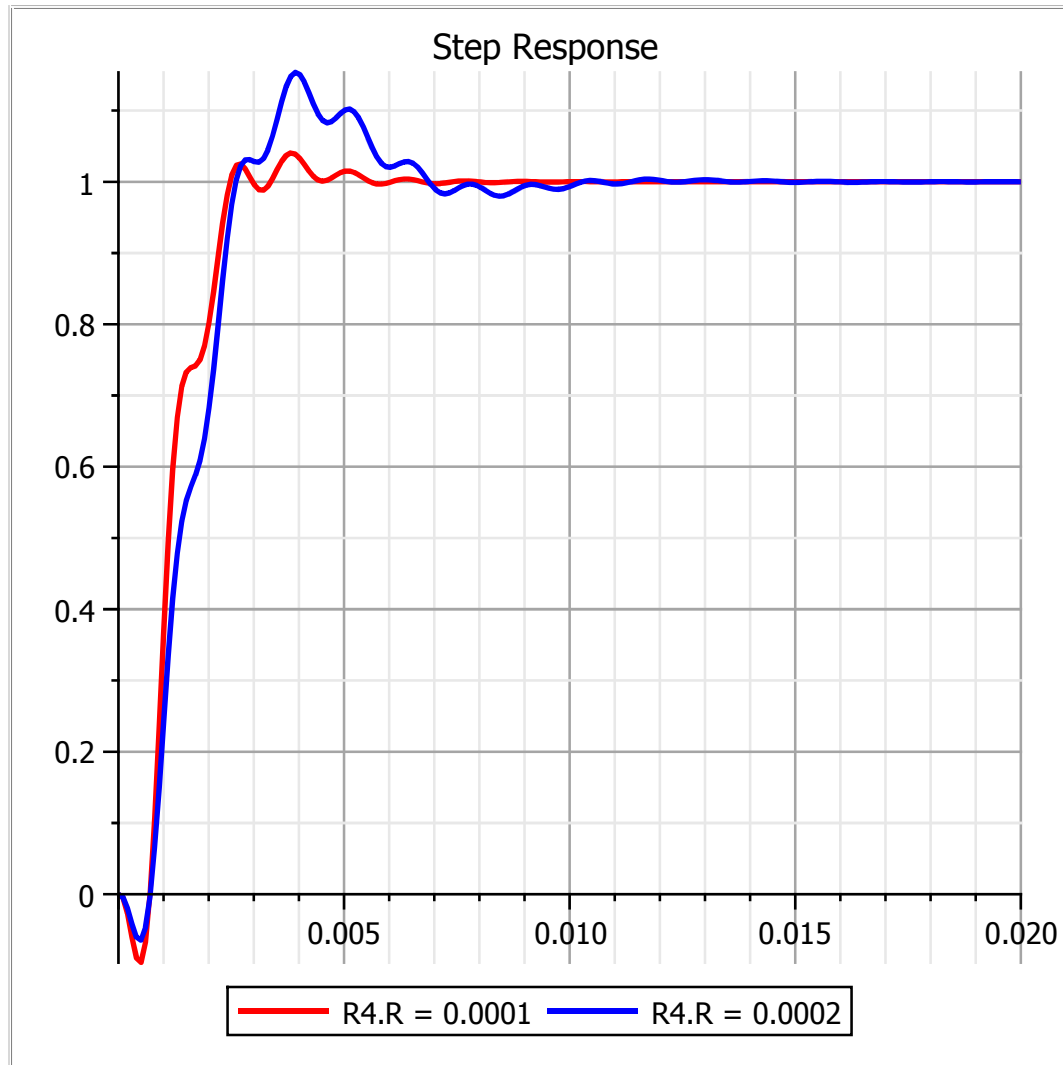
The sensitivity of the system to a change in inductor parameter value is shown in the following step response plot. The inductor parameter value was perturbed from 0.0001 H to 0.0002 H.



The effects of varying the inductor parameter on the system response is negligible.

## ▼ Resistor parameter sensitivity

The sensitivity of the system to a change in resistor parameter value is shown in the following step response plot. The resistor parameter value was perturbed from  $1\Omega$  to  $2\Omega$ .



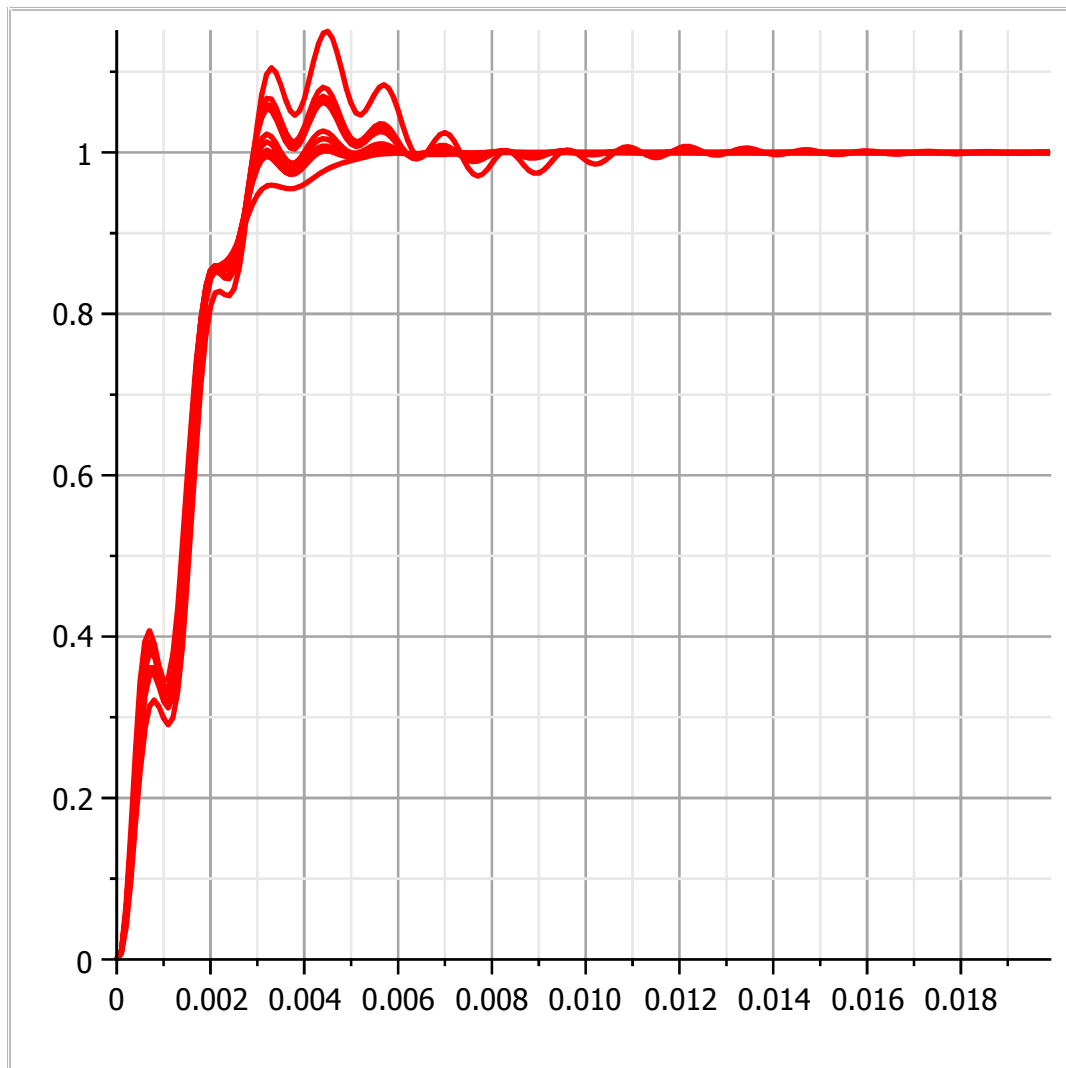
As shown in the plot, increases to the resistor parameter value only affect the amount of overshoot present in the system.

## ▼ Monte-Carlo simulations

As mentioned previously, conducting Monte-Carlo simulations is very useful for understanding the system dynamics of unknown (or uncertain) parameters. The main criterion for conducting Monte-Carlo simulations is that the statistical distribution of the parameter under investigation is known. The table below performs a Monte-Carlo simulation for the EMF, inductor, and resistor parameters with different statistical characteristics.

## Monte-Carlo Simulations

Variable parameter	K <input type="button" value="v"/>	Standard deviation	<input type="text" value="0.2e-1"/>
Mean	<input type="text" value=".1"/>	Number of samples	<input type="text" value="10"/>



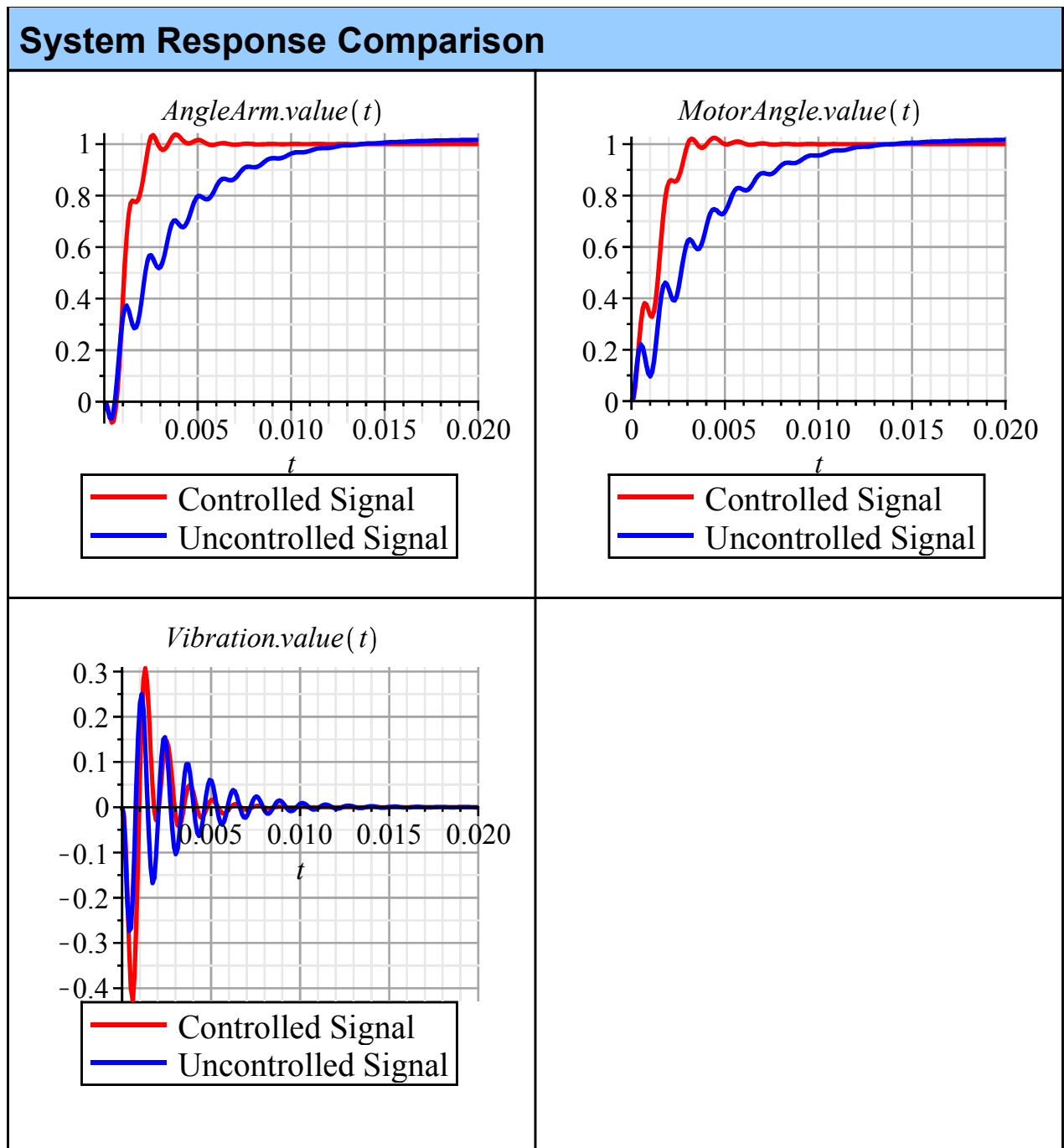
Run Simulation

\*\* Please Note: This simulation takes a few minutes to run.

## ▼ Closed-Loop Response

The equations governing the LQG controller were encapsulated into a MapleSim component using the Custom Component template. (To access the template, refer to the LQRController file located in the Document Folder, or double-click the Controller Model block in the model diagram).

The inclusion of an LQG feedback control system resulted in significant speedup improvements in terms of positioning. This can be verified by comparing the step response of the controlled and uncontrolled system in the plots below. As shown below, the controller removed many of the oscillations that were present in the operating bandwidth of the plant model. This results in a positioning speedup gain of 70% compared to the open-loop response.



## ▼ Results

This document outlines a procedure to quickly and reliably control the position of a voice coil actuator head.

In section 2, the dynamics of the plant model were investigated. The electromagnetic force that was responsible for driving the voice coil actuator was modeled as a simple DC motor, while the voice coil actuator head was modeled by cascading two MapleSim Multibody components, namely a revolute joint and a flexible beam. The DC motor parameters that best describe the behavior of the real-world system were identified by linearizing the nonlinear plant model equations. Using system identification techniques, the DC motor parameters for the inductor and EMF component were found to be 0.0001 H and 0.1 Nm/A respectively. The sensitivity of these parameters, including that of the resistor, was investigated. The open-loop system response showed the plant model to be sluggish in terms of positioning.

In section 3, user-defined algorithms were used to determine the gain values of the LQG controller. The robustness of the controller was investigated by perturbing the DC motor parameter values. As expected, the controller was found to be robust. The equations defining the LQG controller were encapsulated into a custom MapleSim component and simulated with the plant model. The results of the simulation show the closed-loop system to have a 70% speed improvement in comparison to the open-loop system.



[www.maplesoft.com](http://www.maplesoft.com) | [info@maplesoft.com](mailto:info@maplesoft.com)  
Toll-free: (US & Canada) 1-800-267-6583 | Direct: 1-519-747-2373

To view other Application Briefs visit:  
[www.maplesoft.com/appsbriefs](http://www.maplesoft.com/appsbriefs)

Legal Notice: © Maplesoft, a division of Waterloo Maple Inc. 2008. Maplesoft, Maple, and MapleSim are trademarks of Waterloo Maple Inc. All other trademarks are the property of their respective owners. This application may contain errors and Maplesoft is not liable for any damages resulting from the use of this material.