



UCD

/ School of Engineering and  
Architecture

/ Electronic Engineering  
/

# CT 20 1

# Matlab for control

**EEEN40010 Control Theory**

**Authors :** Tiarnach Ó Riada 16315466  
**Date submitted :** 15th November 2020

# Problems

Problem 1 . . . . .	1
Problem 2 . . . . .	2
1 Linearised system model . . . . .	2
2 PID Controller Design . . . . .	2
3 Evaluation of Failure Mechanisms . . . . .	3

## Problem 1

## Problem 2

$$\begin{aligned}
 F(t) &= m \frac{d^2}{dt^2} y + \delta \frac{d}{dt} y + ky + \epsilon y^3 \quad (1) \\
 m &= a + b + c = 16 \\
 \delta &= 5.3c = 31.8 \\
 k &= 0.5(a + 2c) = 8 \\
 \epsilon &: \text{nonlinearity parameter}
 \end{aligned}$$

**Equation 1:** Global model of spring-mass-damper system with  $F(t)$  as input and  $y(t)$  as output.

### 2|1 Linearised system model

$$\begin{aligned}
 y &= 0 \\
 \frac{d}{dt} y &= 0 \quad (2)
 \end{aligned}$$

To linearise eq. (1) about operating point eq. (2) I first replaced the output  $y(t)$  with its operating point value (denoted  $Y_0$ ) and a quantity representing a small deviation about its operating point (denoted  $\tilde{y}$ ) :

$$y(t) = Y_0 + \tilde{y}(t)$$

Subbing this expression into eq. (1) we arrive at the following equation:

$$m \frac{d^2}{dt^2} (Y_0 + \tilde{y}) + \delta \frac{d}{dt} (Y_0 + \tilde{y}) + k(Y_0 + \tilde{y}) + \epsilon(Y_0 + \tilde{y})^3 = F(t)$$

Which—neglecting powers of  $\tilde{y}$  as these would be very small for small deviations from the operating point, neglecting derivatives of  $Y_0$  as  $Y_0$  is constant, and neglecting  $Y_0$  as  $Y_0 = 0$ —becomes the following:

$$m \frac{d^2}{dt^2} \tilde{y} + \delta \frac{d}{dt} \tilde{y} + k\tilde{y} = F(t) \quad (3)$$

Let the elements of the state vector  $\mathbf{x}$  be the deviation of the deflection  $\tilde{y}$  and its first derivative  $\frac{d}{dt} \tilde{y}$ :

$$\mathbf{x} = \begin{bmatrix} \tilde{y} \\ \frac{d}{dt} \tilde{y} \end{bmatrix} \quad (4)$$

To find an expression for the derivative of the state vector  $\dot{\mathbf{x}}$ , equations are derived as follows:

$$\begin{aligned}
 \frac{d}{dt} x_1 &= x_2 \\
 \frac{d}{dt} x_2 &= \frac{1}{m} (F(t) - \delta x_2 - k\tilde{y})
 \end{aligned}$$

Representing the input  $F(t)$  by its operating point value of  $F_0 = 0$  and its small deviation from this value by  $\tilde{F}$  we arrive at the following representation of  $\dot{\mathbf{x}}$ :

$$\begin{aligned}
 \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\delta}{m} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \tilde{F} \quad (5) \\
 &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\
 \mathbf{u} &= [\tilde{F}] : \text{input vector}
 \end{aligned}$$

As the output  $y$  in this model is just the deviation  $\tilde{y}$  the output is given by the following equation:

$$y = [10] \mathbf{x} + [0] \mathbf{u}$$

The transfer function of this local LTI model was calculated in MATLAB using the `ss` and `tf` commands:

$$\mathcal{G}_{p,lin} = \frac{0.0625}{s^2 + 1.987s + 0.5} \quad (6)$$

The characteristics of the system's response to a step input are given in table 1.

Characteristic	Value
Steady-state error	0.8752
$t_{s,2\%}$	13.8878 s
PO%	0

**Table 1:** Step characteristics of the local LTI model of the plant eq. (6).

### 2|2 PID Controller Design

The implications of the design specifications are given below:

- Closed loop system must be stable.** All poles and zeroes of the closed loop system must have negative real part.
- Zero steady-state error to a step input.** The open-loop system must have a pole at zero. This implies the presence of an I term in the controller as the plant does not have a pole at zero.
- $t_{s,2\%} \leq 60\% t_{s,2\%,plant}$ . The settling time must be less than 8.3327 s. Approximating as a first-order system, this requirement implies a pole at  $\sigma < -\frac{4}{t_s} = -0.48$ .
- Percentage overshoot less than 50% that of the plant or 25%, whichever is greater.** The plant's overshoot is 0, so the controller must ensure that the closed loop system exhibits less than 25% overshoot.

**First I tried a PI controller**, as these were thus far the only necessary terms. The equation for this controller is given below:

$$\mathcal{G}_c = \frac{k(s+z)}{s} \quad (7)$$

I chose  $z$  to be 0.5 to satisfy the settling time requirement. Using `rlocfind` I calculated the gain for there to be a dominant pole at  $\sigma = -0.7$  in the open-loop system. I chose this value as it seemed reasonably greater in magnitude than the required  $\sigma = -0.48$  to give some margin for errors in the assumption governing the settling time requirement. I chose a real pole as introducing a complex pair would have added oscillations unnecessarily into the system. The value of gain returned was  $\sim 23$ .

The final controller equation is the following:

$$\mathcal{G}_c(s) = \frac{23(s + 0.5)}{s} \quad (8)$$

The step response of the closed loop system is given in fig. 2.1 and the associated characteristics in table 2. As the table shows, the system is stable (the real part of all poles is negative), the steady state error is zero (final value is one), the 2% settling time is less than the requisite 60% of the plant's settling time given in table 1, and the percentage overshoot is less than the requisite 25%.

Characteristic	Value
Poles	$-0.3941 \pm 0.4038i, -1.1994$
Zeros	$-0.5$
Steady-state error	0
$t_{s,2\%}$	6.1386 s
PO%	18.3072%

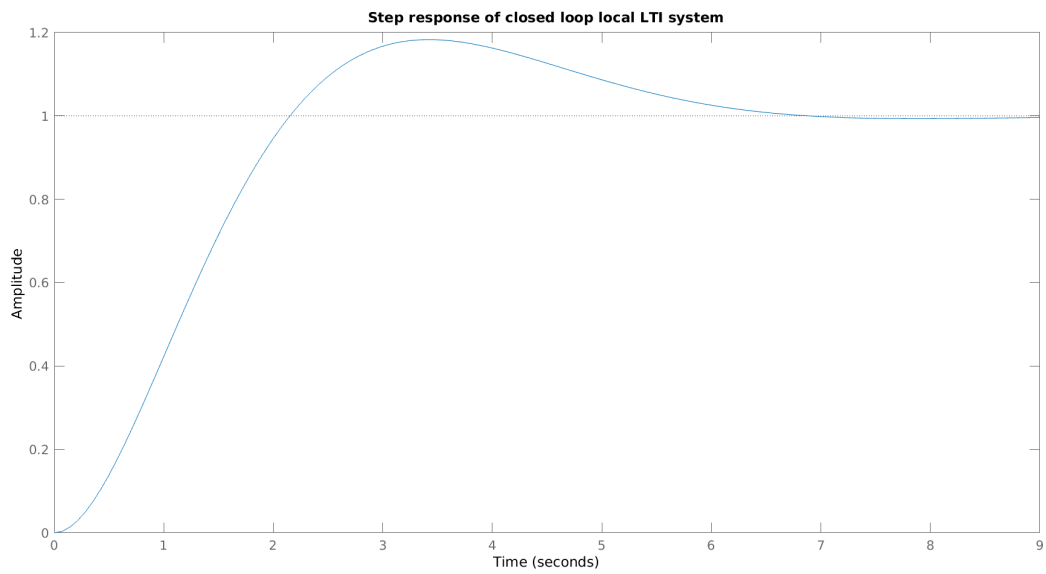
**Table 2:** Step characteristics of the closed loop system of  $G_c$  in series with  $G_{p,lin}$  with ideal feedback.

### 2|3 Evaluation of Failure Mechanisms

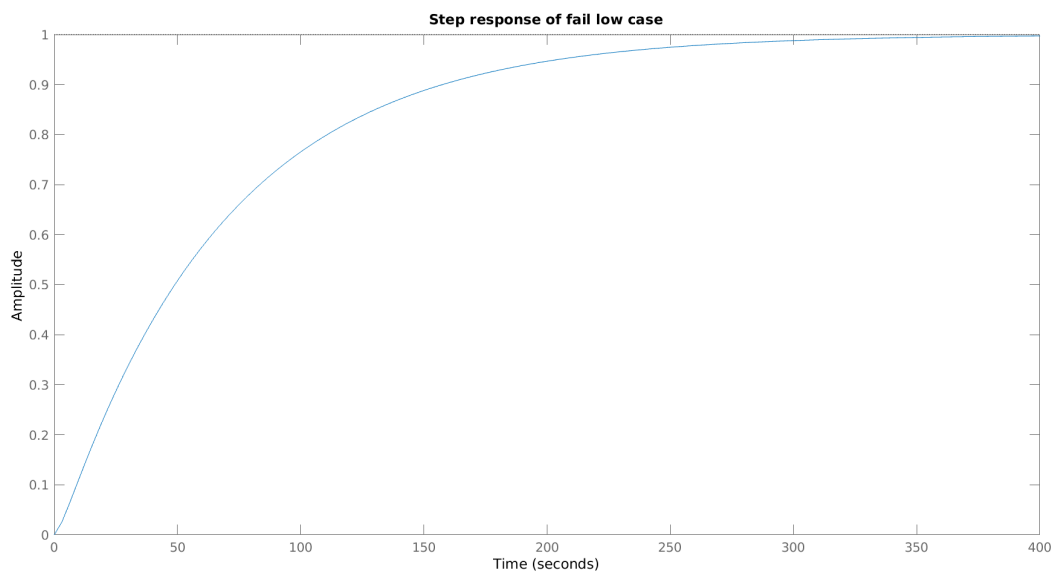
For the case where the gain fails to 1% of the nominal, the step response is shown in fig. 2.2. As the plot shows, the system is extremely slow ( $t_s = 266$  s), but exhibits no overshoot. This lack of overshoot implies that the system is failsafe, unless a failure condition is extremely sluggish response.

For the case where the gain fails to 500% of its nominal value, the step response is shown in fig. 2.3. Contrary to the previous failure case, the 2% settling time of the step response of this failure mode is not too dissimilar to the nominal value. Its overshoot is 49% however, nearly twice the specified value. This is most likely unsafe.

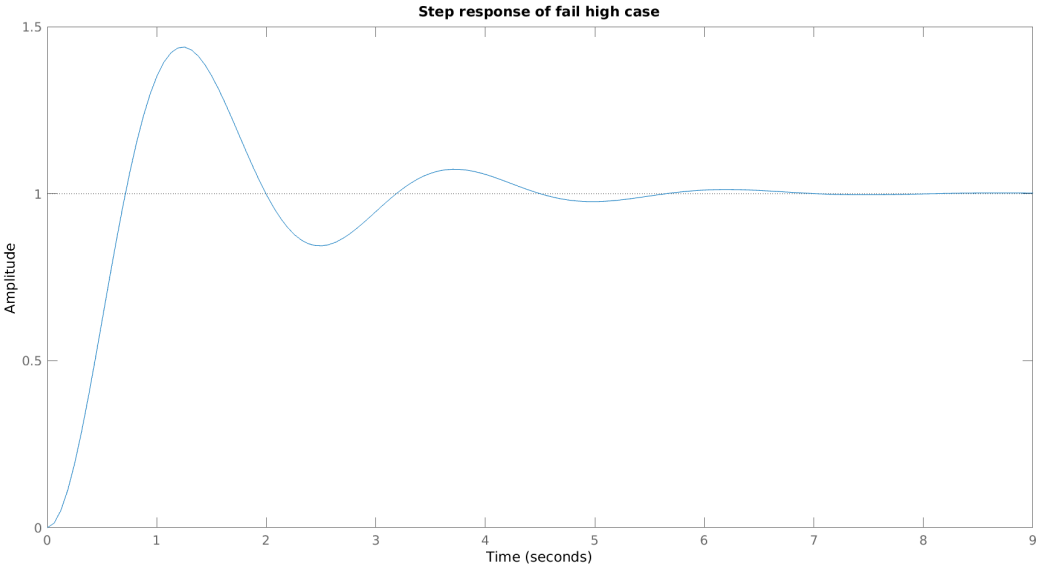
Judged by these two failure modes, the closed-loop system is not failsafe due to the magnitude of the overshoot exhibited by the fail-high mechanism and possibly also the extreme slowness exhibited by the fail-low mechanism.



**Figure 2.1:** Step response of closed loop system of  $G_c$  in series with  $G_{p,lin}$  with ideal feedback.



**Figure 2.2:** Step response of fail low scenario.



**Figure 2.3:** Step response of fail high scenario.