

# **Control Challenges: Solutions**

Iacopo Moles

2025-07-07

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What is this? . . . . .	3
<b>2</b>	<b>Block With Friction</b>	<b>4</b>
2.1	State Space representation . . . . .	4
2.2	Pole Placement . . . . .	6
2.3	Simulation . . . . .	7

# 1 Introduction

## 1.1 What is this?

This is a collection of write ups on how to solve the various problems presented by [Github user](#) “Janismac”.

## 2 Block With Friction

### 2.1 State Space representation

We can convert the set of ODE into a state space representation. The final bode plot of the block position is: Figure 2.1

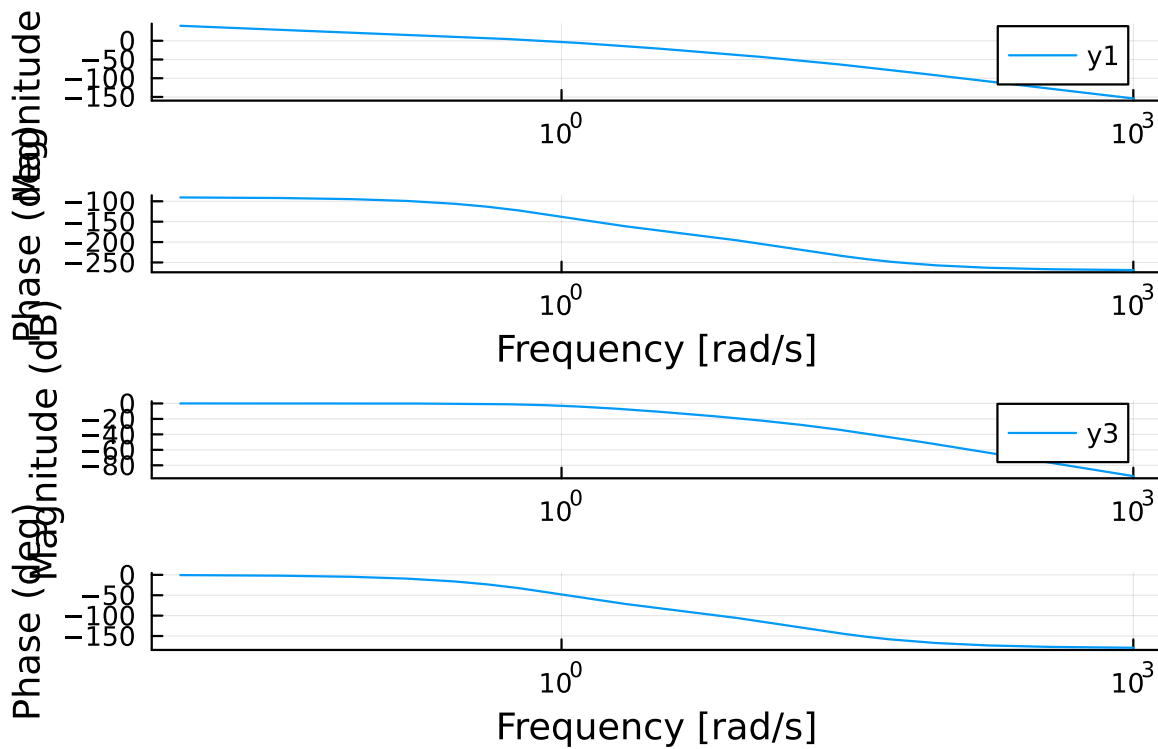


Figure 2.1: Starting Bode Plot

It has the shape we expect from a motor + friction. Slow pole for the mass + friction and a faster pole for the current & inductance.

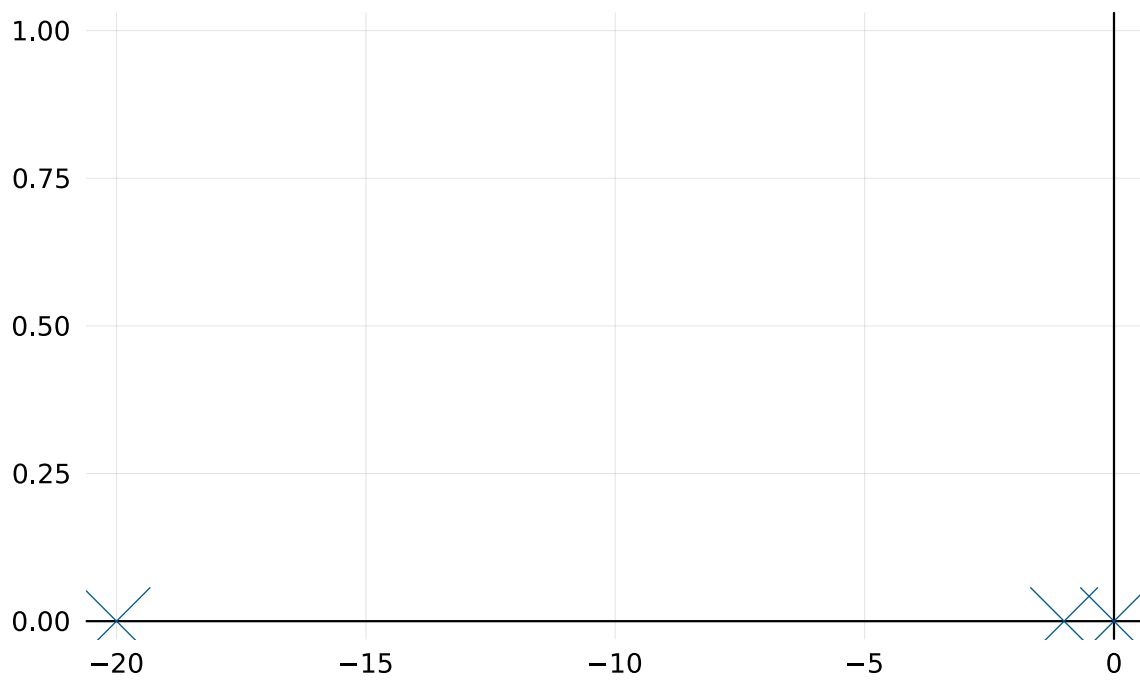
Numerically they are:

In Figure 2.2 we see that we start with all the pole in the left-half plane, which is good.

```
3-element Vector{Float64}:  
-20.0  
-1.0  
0.0
```

(a) Starting PZ map

## Pole-zero map



(b)

Figure 2.2

## 2.2 Pole Placement

We can design a controller with pole placement.

For some reason pole placement doesn't work for the observer, I use a Kalman Filter with random fast values.

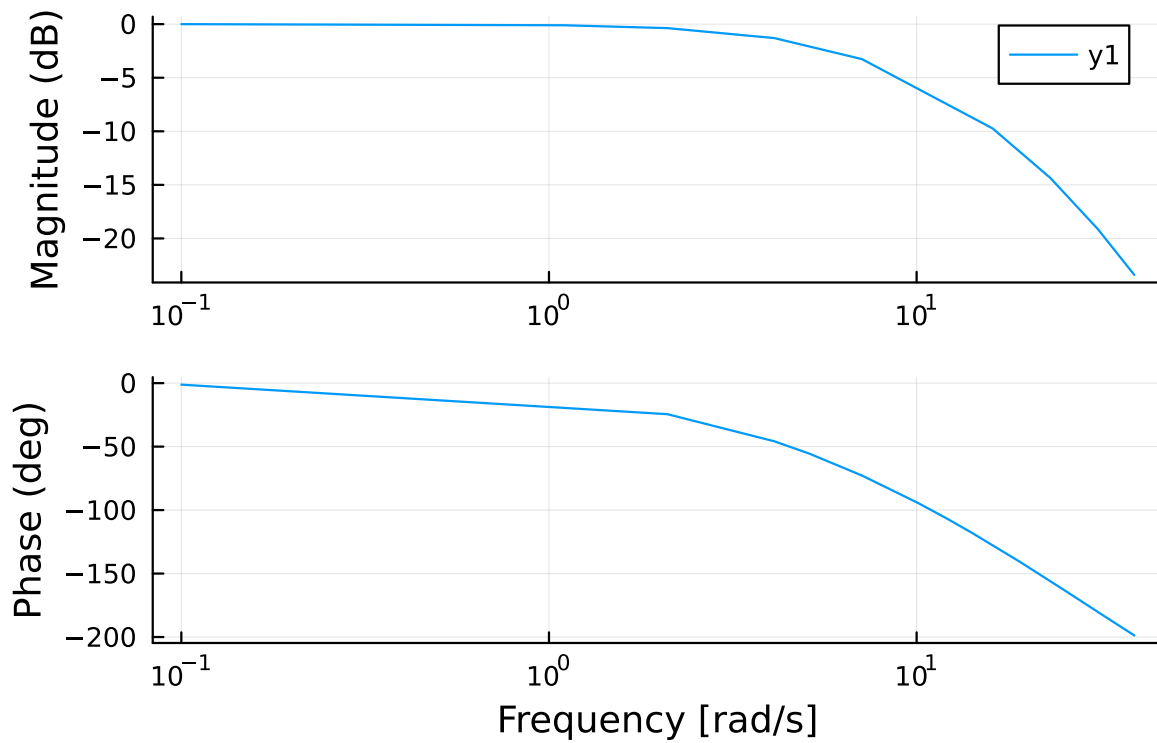
```
(isobservable = true, ranks = [3, 3, 3], sigma_min = [0.05255163155979671, 1.0000000000000002,
```

```
Warning: Max iterations reached
```

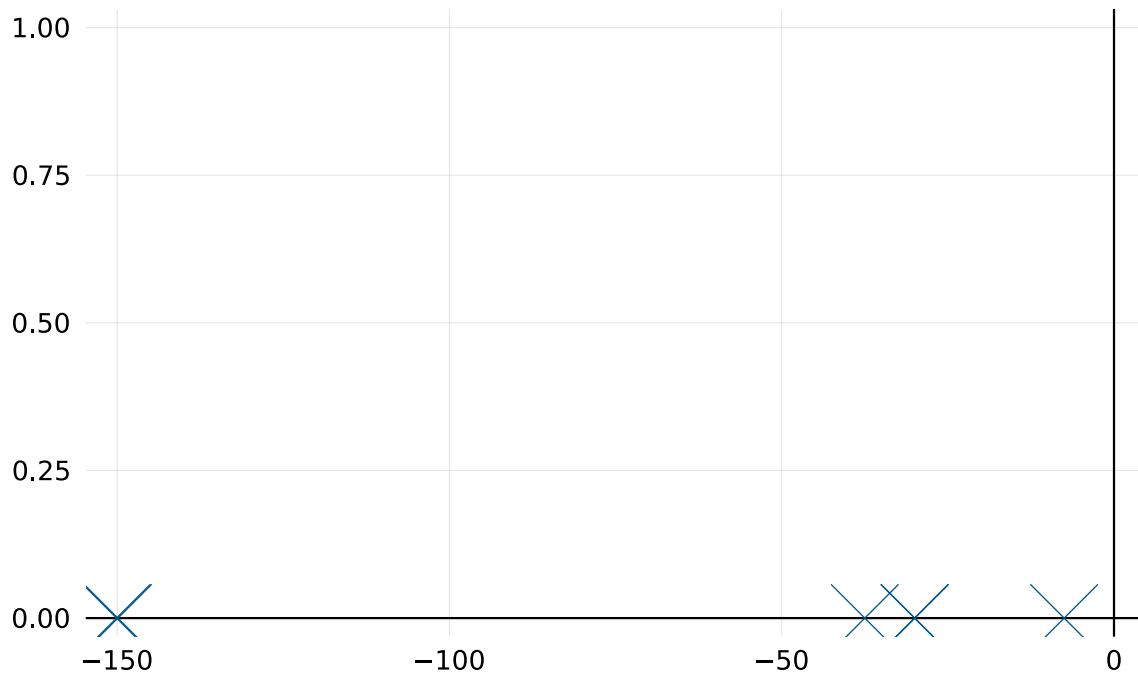
```
@ ControlSystemsBase C:\Users\icpmoles\.julia\packages\ControlSystemsBase\IeuPW\src\synthes
```

We can check the effect of the new controller on the loop

```
ComplexF64[-150.09999999999997 + 0.0im, -149.89999999999995 + 0.0im, -7.500000000000134 + 0.0im]
```



## Pole-zero map

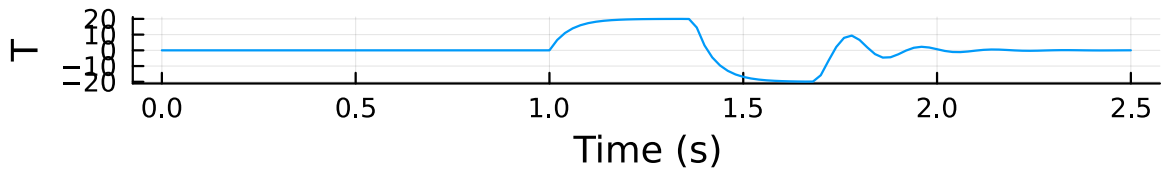
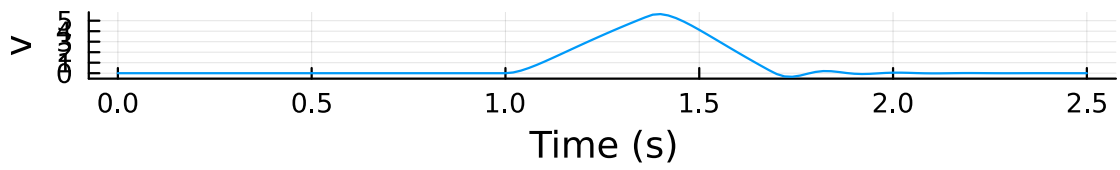
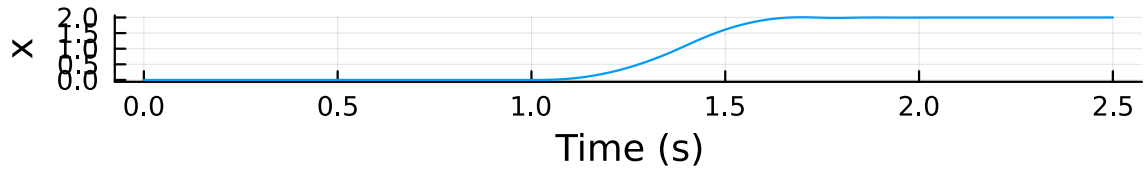
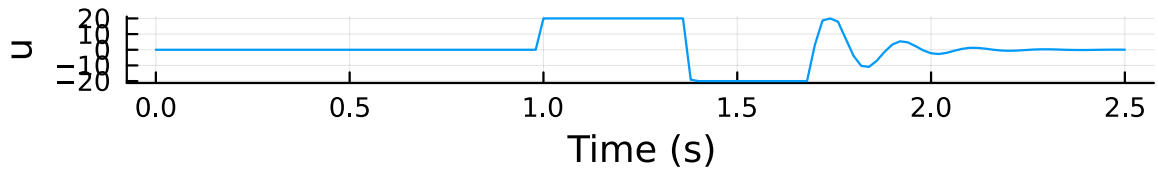


From what I understand we are interested in the dotted line in the bottom right. See how flat it is.

We can convert it to the standard PD gain form.

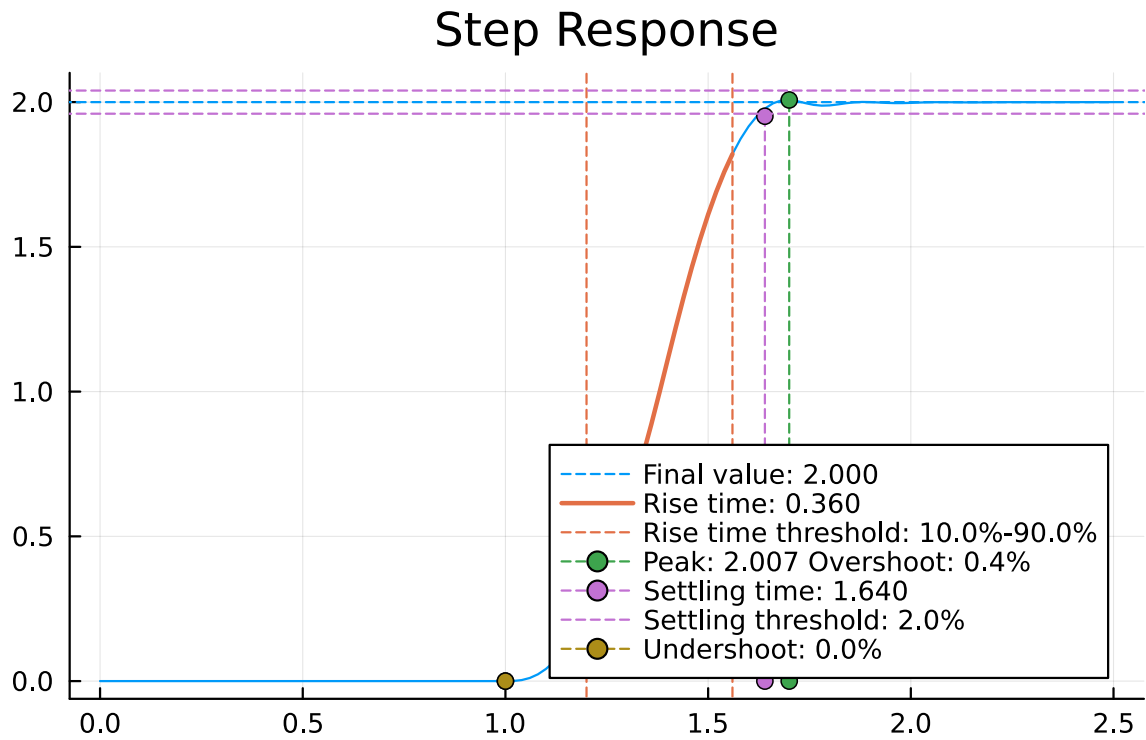
### 2.3 Simulation

We can simulate this with a motor that only outputs the position:

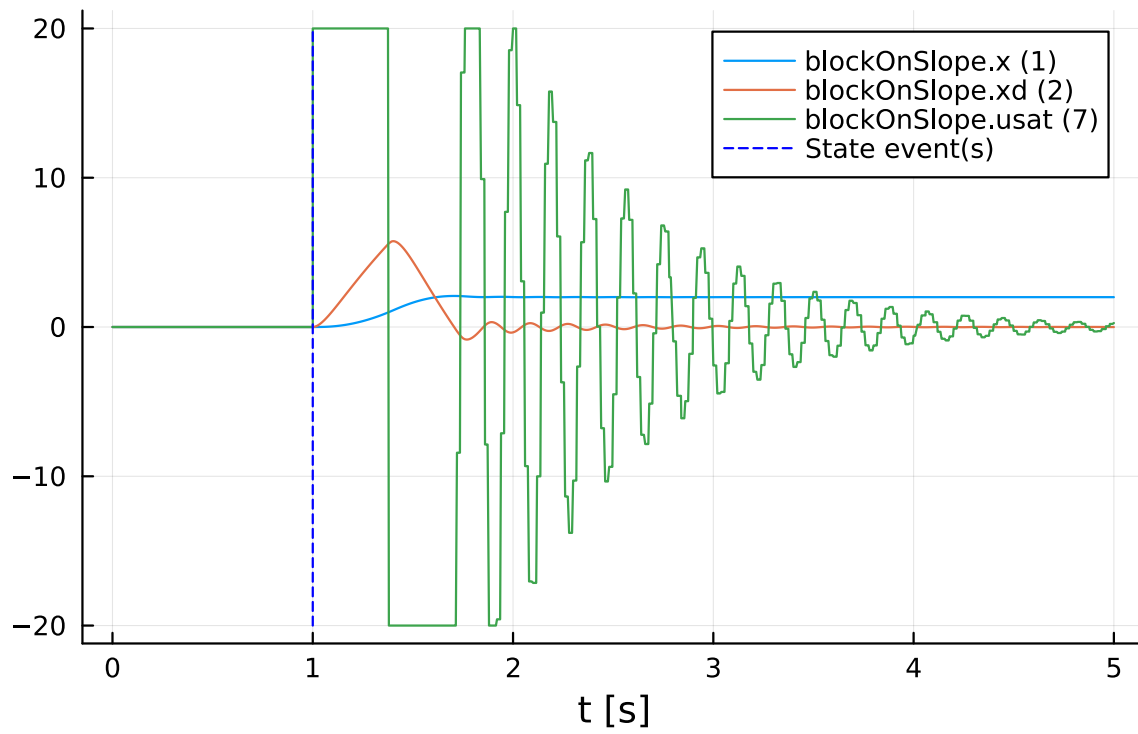


For more stats:





We can also simulate it in a SIMULINK-like environment:



There is a slight difference between the `lsim` simulation and the FMU simulation. I need to recheck some stuff.