COMP431, ASSIGNMENT 2

Hague, Benjamin

DECEMBER 13, 2019

THE UNIVERSITY OF LIVERPOOL

Contents

Part 1. Modelling of DC Motor

Firstly, a motor must be modelled using Equation 1 and Equation 2, to define the characteristics of the DC motor

Equation 1, Current Equation for DC motor

$$V_a - E_a = R_a I_a + L_a \frac{dI}{dt}$$

Equation 2, Torque Equation for DC motor

$$T_e = T_l + J\frac{d\omega_m}{dt} + b\omega_m$$

Equation 3, Relationship between Torque and Current in a DC motor system

$$T_e = K_t I_a$$

Equation 4, Relationship between Current and Rotational Velocity in a DC motor system

$$E_a = K_e \omega_m$$

Where

Va: Armature voltage (V)

Te: Electronically generated torque (Nm)

TL: Torque of the mechanical load (Nm)

J: The moment of inertia (kg m2)

We must first rearrange Equation 1 to give Equation 5 and Equation 2 to give Equation 6. The relationship between these equations is defined by Equation 3 and Equation 4.

Equation 5, Rearranged Characteristics for current in DC motor system

$$\frac{dI}{dt} = \frac{V_a - E_a - R_a I_a}{L_a}$$

Equation 6, Rearranged characteristics for Torque in a DC motor system

$$\frac{d\omega_m}{dt} = \frac{T_e - T_L - b\omega_m}{J}$$

Equation 5 and Equation 6 are in a form which can be modelled within MatLab Simulink as shown by Figure 1. We can see each section separately within this figure. The upper half of the image represents the current I_a whereas the lower section represents the rotational speed ω_m we can observe the relationship between the variables is given by the connection which occurs through the gain blocks labelled Ke and Kt.

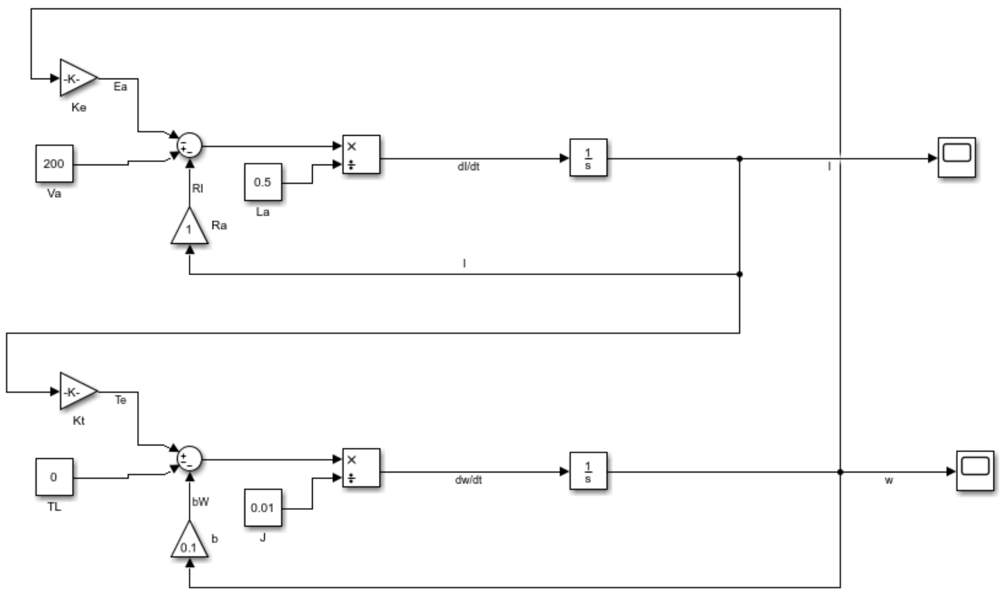


Figure 1, MATLAB Simulink Model for Equation 6

We can observe similar waveform shapes occurring in Figure 2, Figure 3 and Figure 4. Each waveform corresponds to the motor characteristic during a no-load simulation for speed, current and Torque respectively.

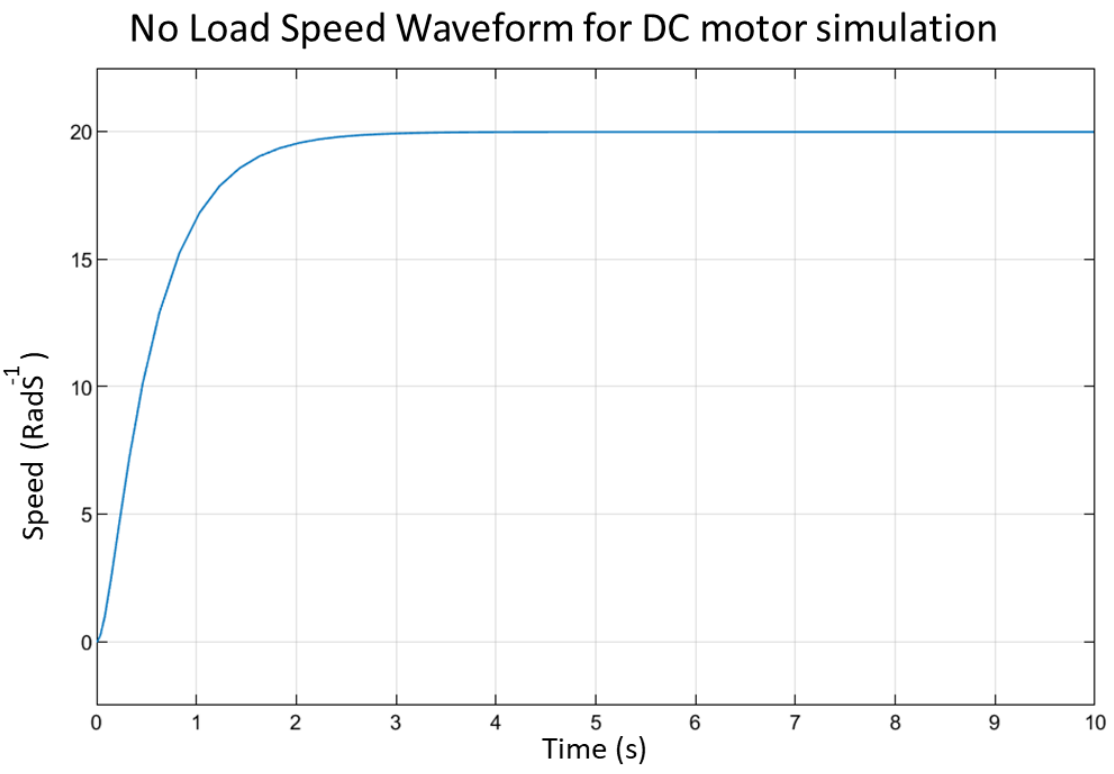


Figure 2, Graph Showing the Speed of a DC Motor with no Load over the first 10 seconds of operation

Figure 2 shows the maximum no-load speed for this motor is 20 RadS^{-1} this is as we would expect.

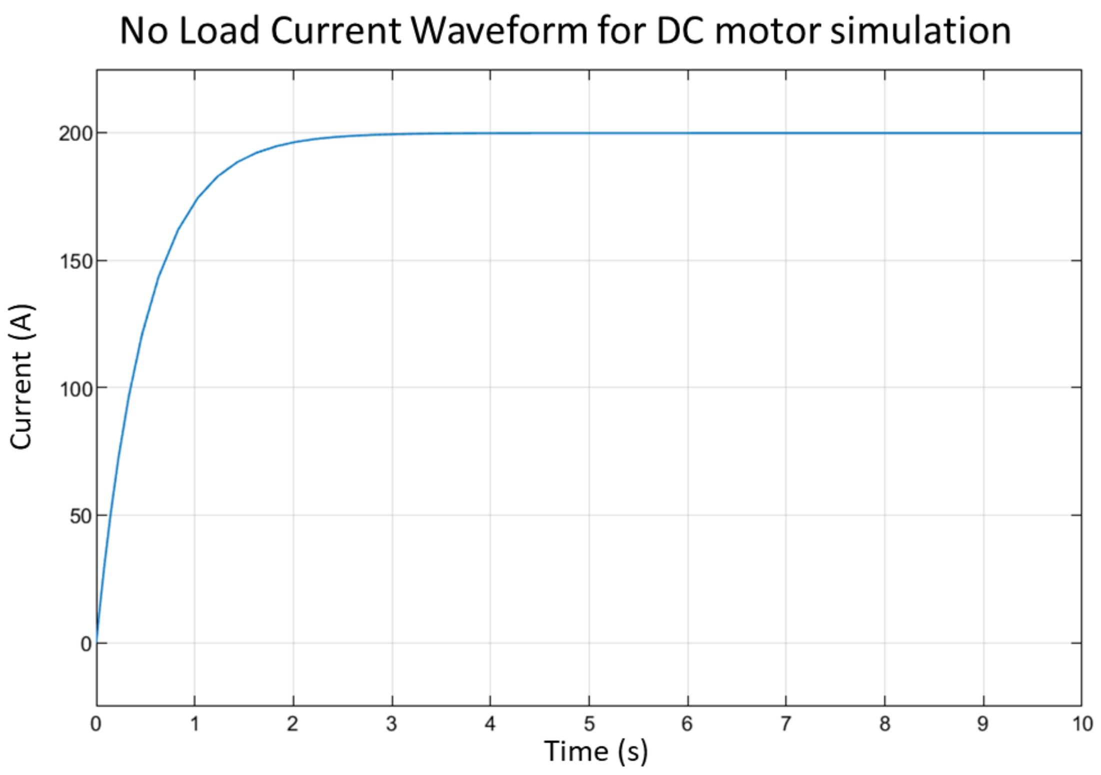


Figure 3, Graph showing the Current Drawn from a simulated DC motor over the first 10 seconds of operation

Figure 3 shows the maximum no-load current for this motor is 200A. This is as we would expect for this style of system. Figure 4 shows the generated torque for this motor is 2Nm. This is also as we would expect.

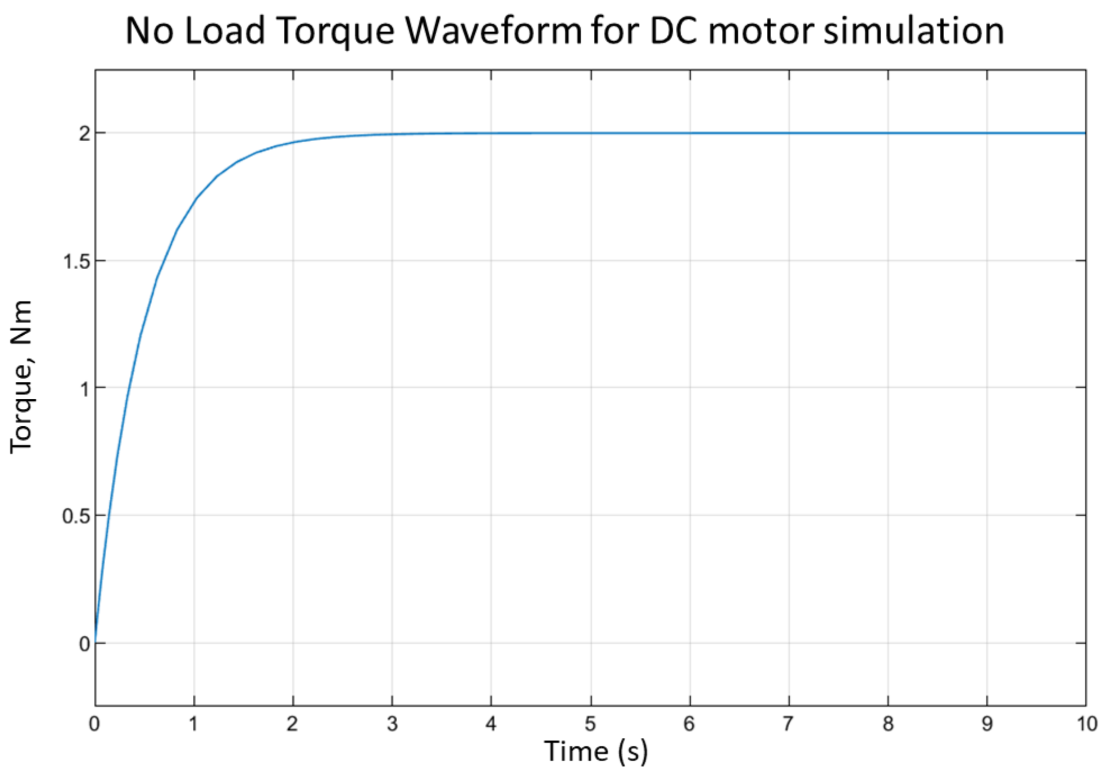


Figure 4, Graph showing the electrical torque produced by a simulated DC motor over the first 10 seconds of operation

Part 2. Model Verification

Model verification has been achieved using a DC machine in the Simulink software.

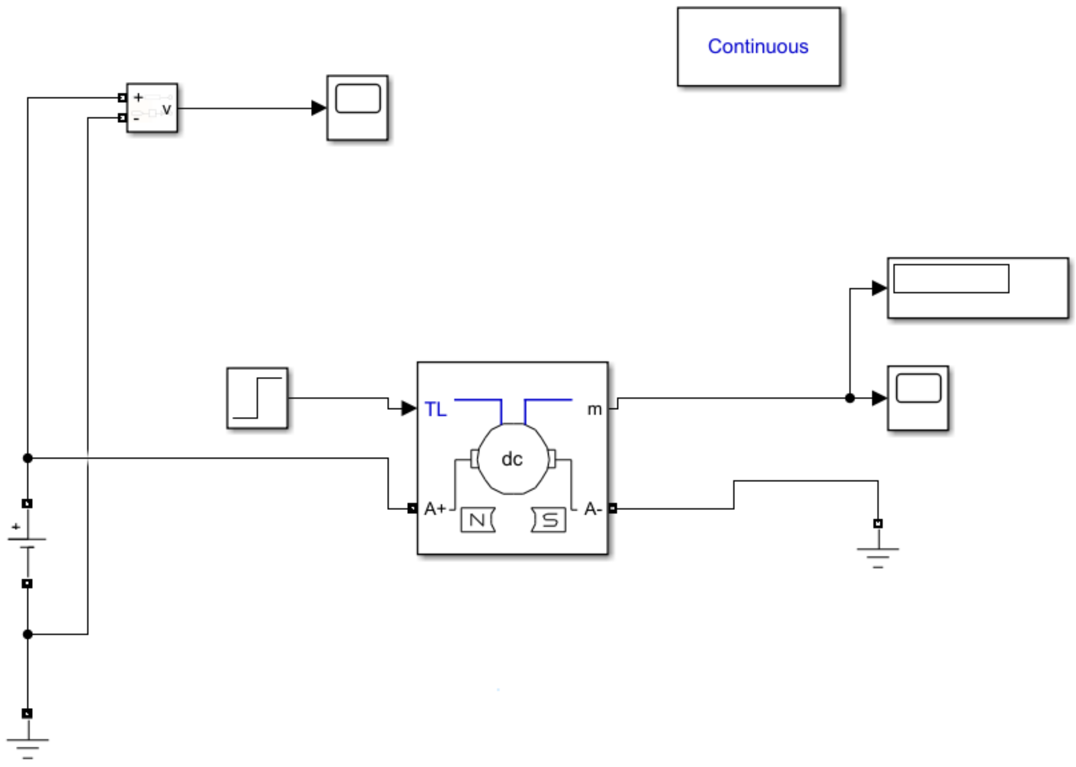
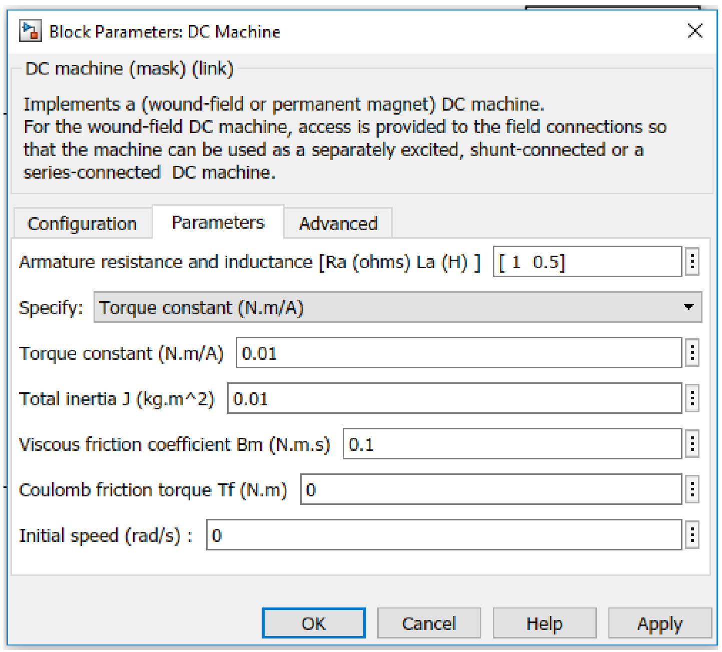


Figure 5, DC Machine Block Diagram

Figure 5 shows the full Block diagram for this setup

Figure 6 evaluates the configuration of the DC machine with a step input of 0 to represent the load on the motor.



Block Parameters: DC Machine

DC machine (mask) (link)

Implements a (wound-field or permanent magnet) DC machine.
For the wound-field DC machine, access is provided to the field connections so that the machine can be used as a separately excited, shunt-connected or a series-connected DC machine.

Armature resistance and inductance [Ra (ohms) La (H)] [1 0.5]

Specify: Torque constant (N.m/A)

Torque constant (N.m/A) 0.01

Total inertia J (kg.m^2) 0.01

Viscous friction coefficient Bm (N.m.s) 0.1

Coulomb friction torque Tf (N.m) 0

Initial speed (rad/s) : 0

Figure 6, DC Machine configuration

The results of the simulation are shown in Figure 7, where we can evaluate that the Current is 200A, Speed is 20 RadS^{-1}, and the torque is 2nM.

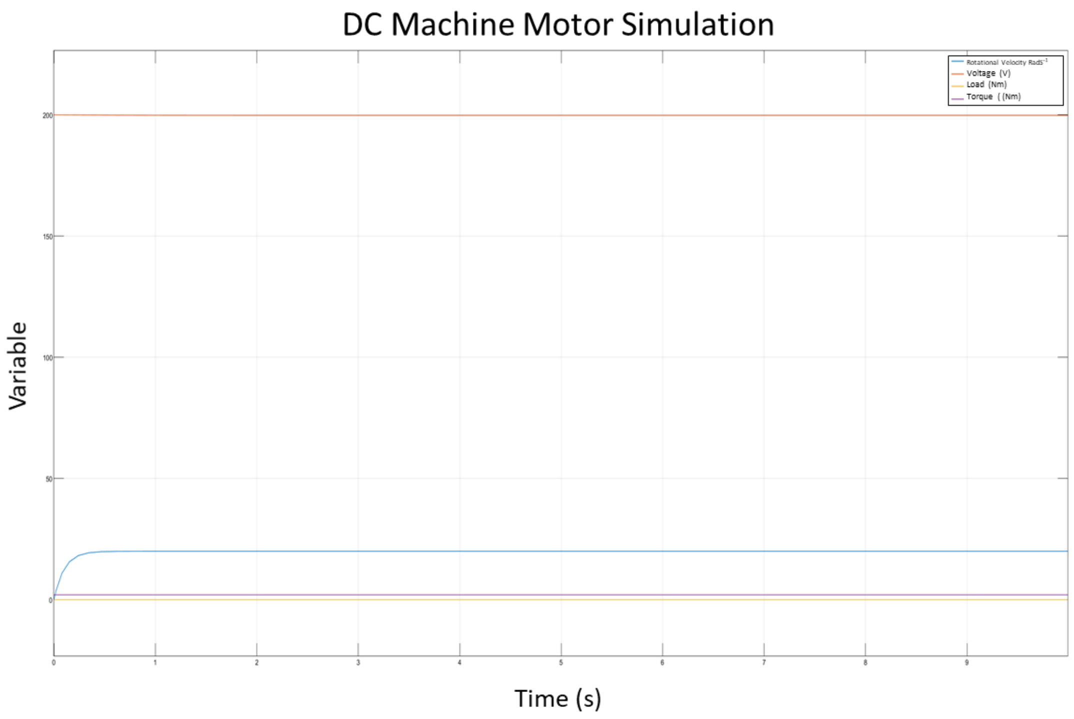


Figure 7, Graph from DC machine, 200 – current, 20 - speed, 2 - Te, 0 – Load

Part 3. Speed Regulation

In order to provide consistent control over a system, a feedback loop is used with a PID controller.

A PID controller provides real-time feedback-driven control to a system. The Premise of the PID controller uses Equation 7 with set values for P, I and D.

Equation 7, PID Control equation

$$P + \frac{I}{s} + \frac{DN}{1 + \frac{N}{S}}$$

We set the reference value in the step block to 20 RadS^{-1} this means out system will vary the input voltage to maintain a steady speed of 20 RadS^{-1}. The defined feedback loop is shown in Figure 8. Figure 9 Builds on this to provide monitoring instruments for the Input speed and the current speed given by the motor. This allows us to predict response times accurately and evaluate the performance of our feedback loop.

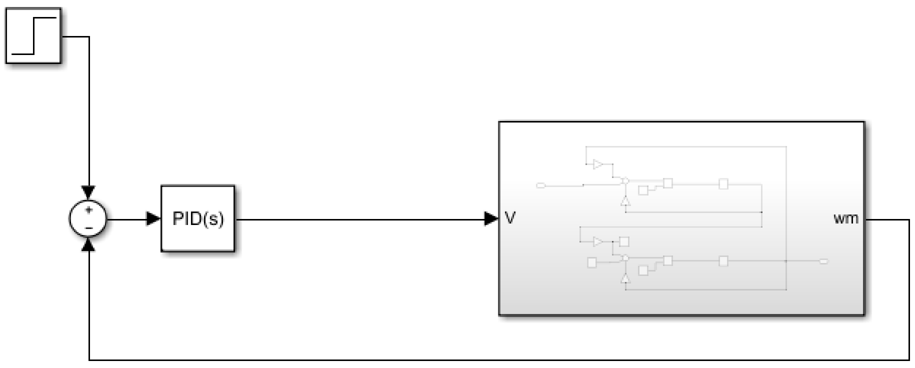


Figure 8, PID Controlled motor model

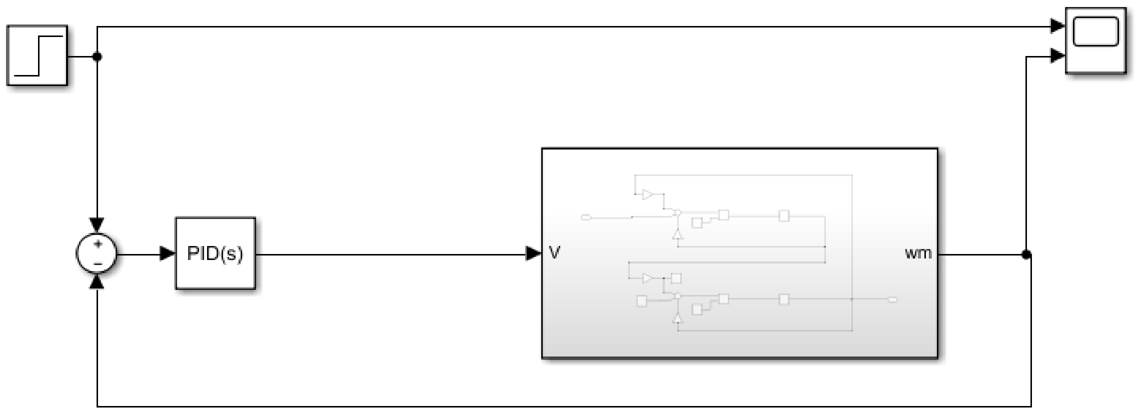


Figure 9, PID controlled motor model with scope output for current speed and expected speed

Through experimentation, we can see how different proportional and integral values affect the settling time and overshoot. Figure 10 shows the reaction to a system where P = 50, I = 50 and D = 0. We can observe that this system achieves a settling time, within 5% of the nominal value, of 0.85s. we can compare this to the system where P = 100, I = 50 and D = 0, as shown in

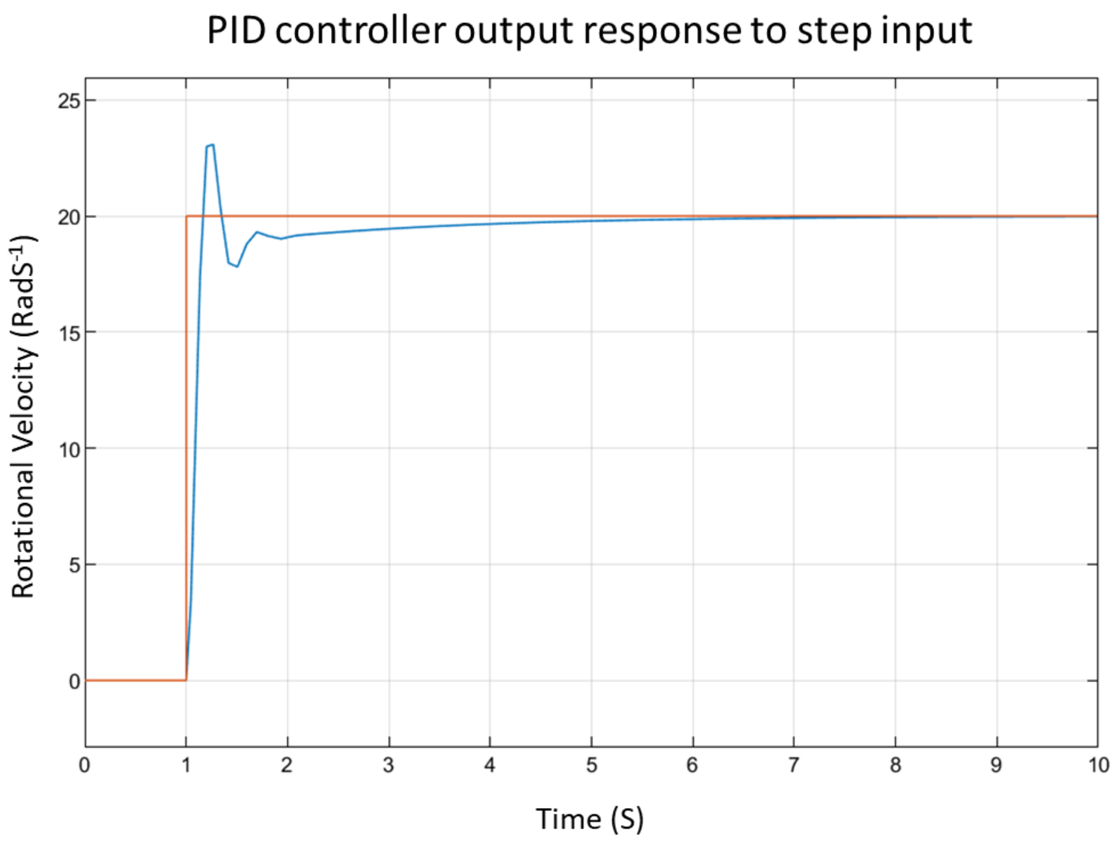


Figure 11. Whilst the settling time for this system is quicker (0.6s), we can also observe that the overshoot is much larger. This is the trade-off we must account for when designing the system.

We can make a similar comparison by increasing the value of I to 200; we can then see a faster settling time (0.56s). but a much higher overshoot of over 25%.

So far within this tuning, we have ignored the derivative option within the controller. This slows the rate of change and therefore increases settling time and overshoot. Figure 13 shows the introduction of a Derivative element.

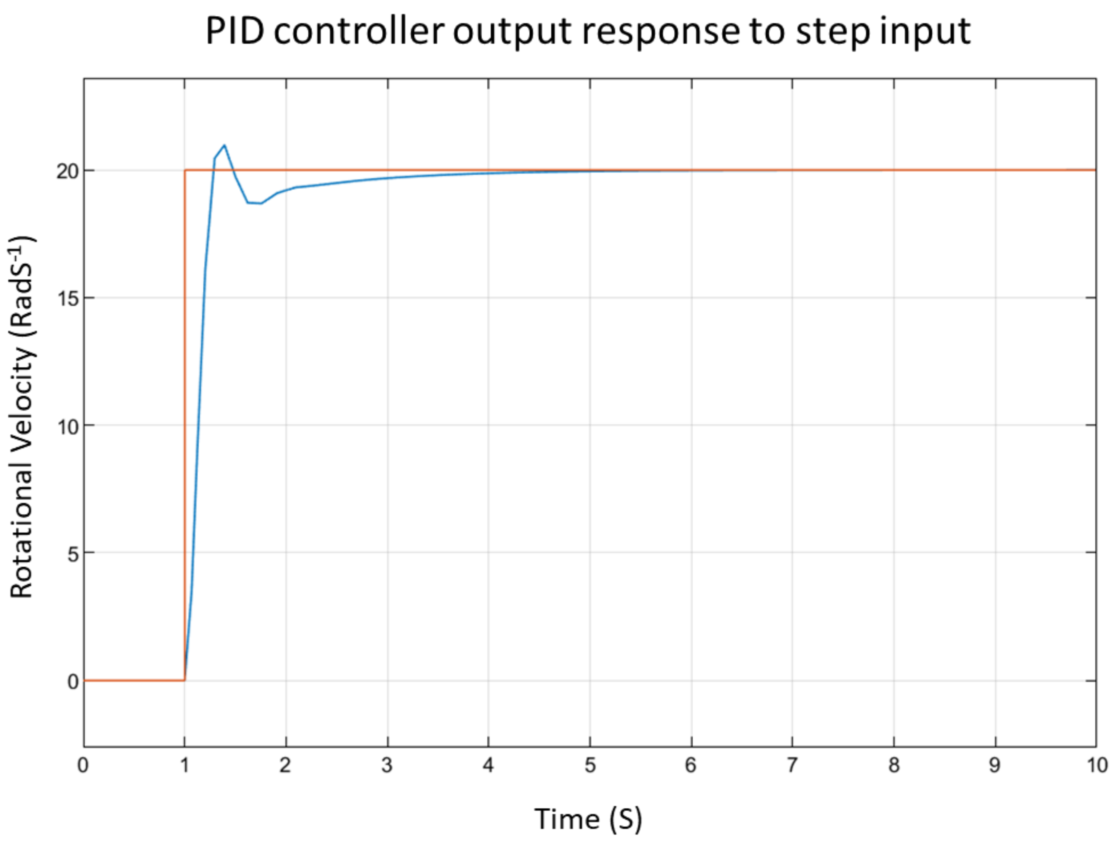


Figure 10, Graph from PID controlled motor where P= 50, I = 50 and D = 0

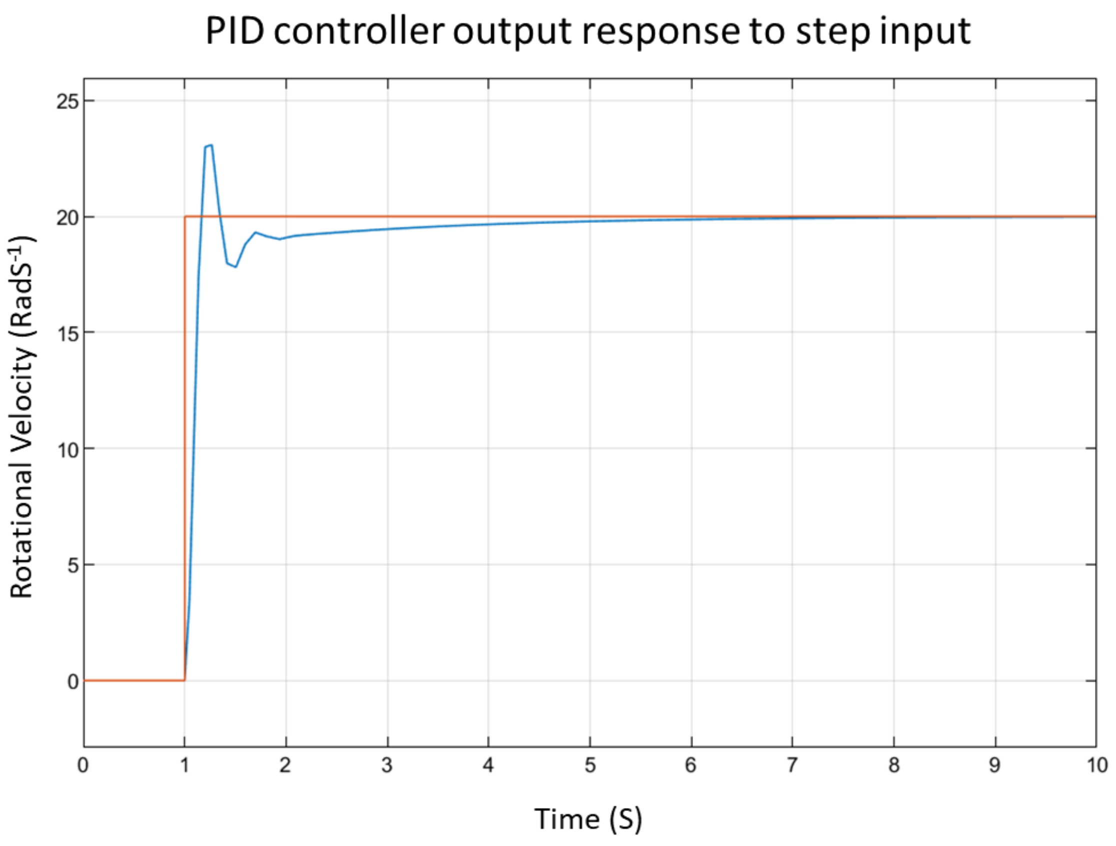


Figure 11, Graph from PID controlled motor where P = 100, I = 50 and D = 0

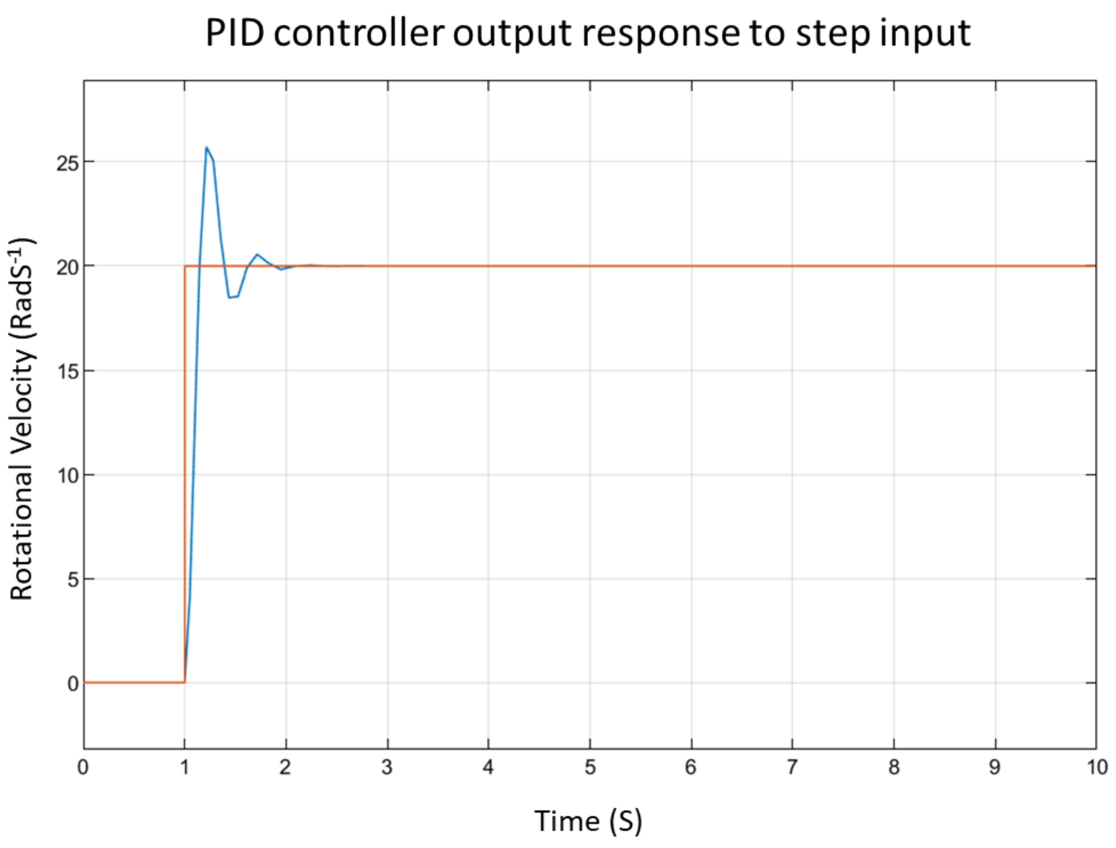


Figure 12, Graph from PID controlled motor where P = 100,I = 200 and D = 0

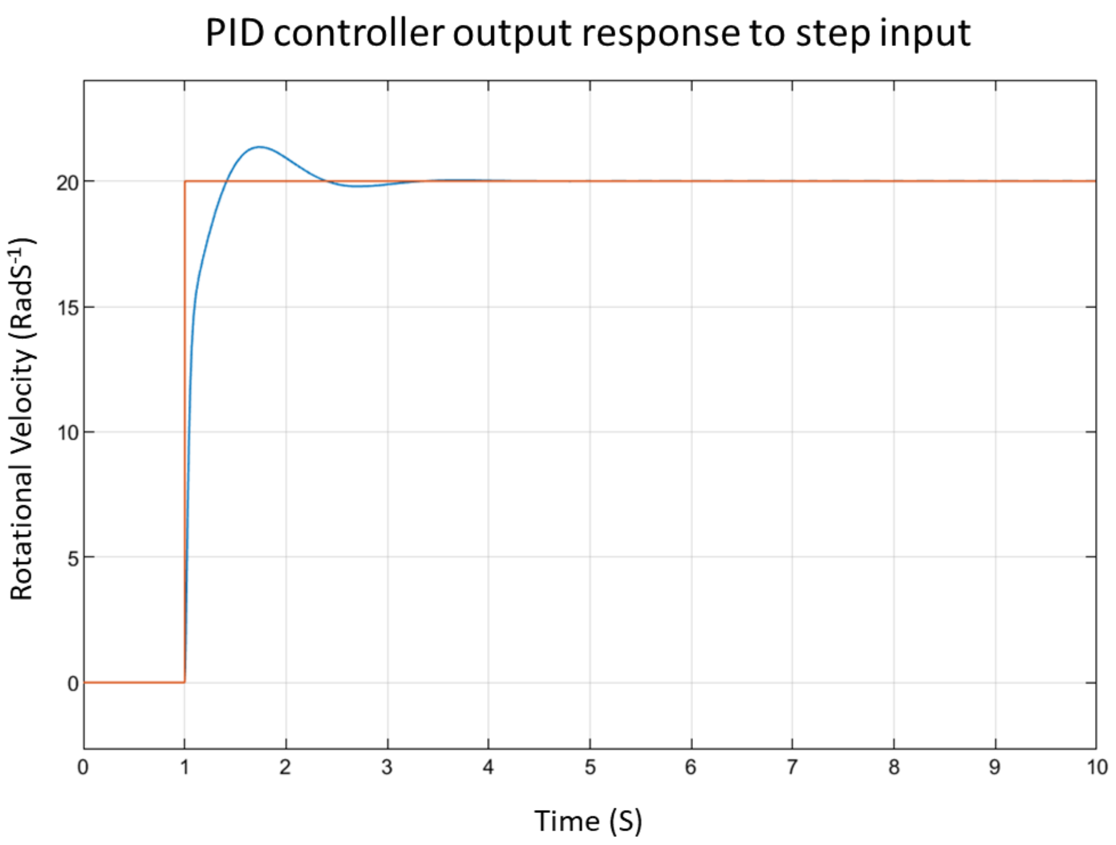
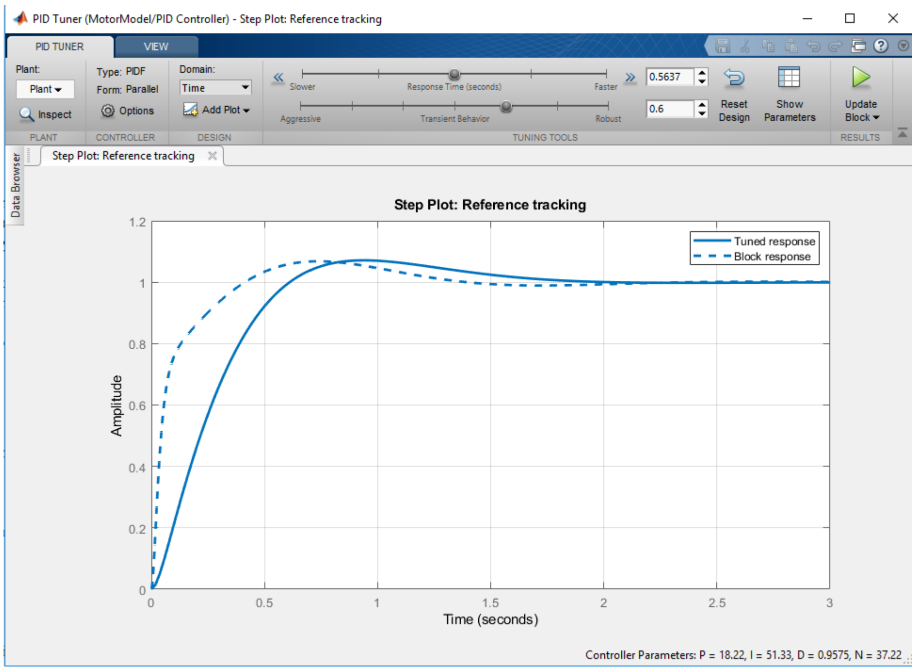


Figure 13, Graph from PID controlled motor where P = 100, I = 200 and D = 10

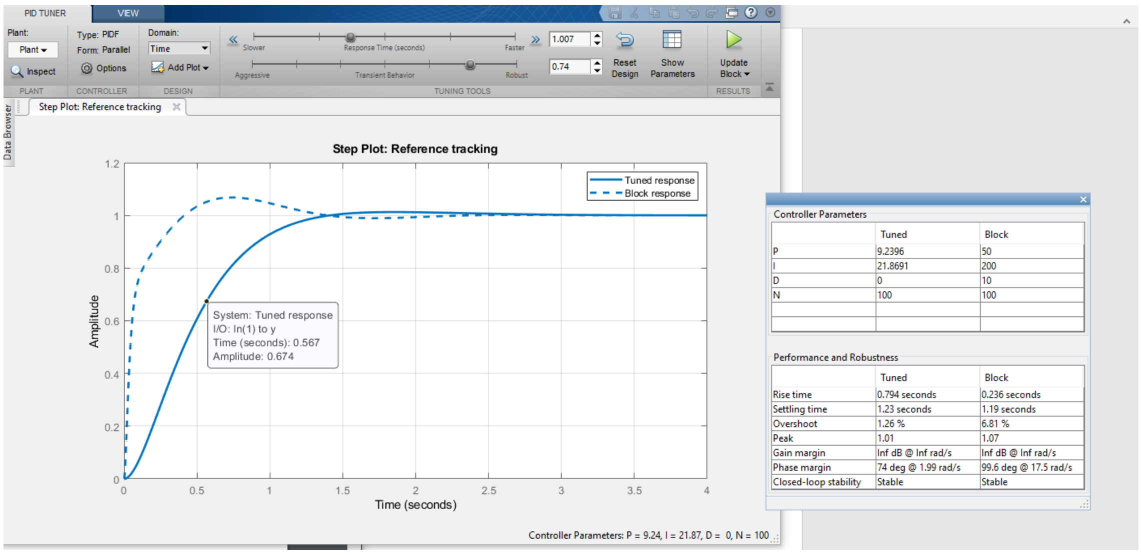
Whilst within the Simulink software, we can manually define parameters to define the PID characteristics. Simulink offers a Direct interface for tuning the parameters based on response time and the transient response of the system. This interface is shown in Figure 14.


PID Tuner (MotorModel/PID Controller) - Step Plot: Reference tracking

Type: PIDF
Form: Parallel
Domain: Time
Response Time (seconds): 0.5637
Transient Behavior: 0.6

Controller Parameters: P = 18.22, I = 51.33, D = 0.9575, N = 37.22

Figure 14, PID tuning window



Type: PIDF
Form: Parallel
Domain: Time
Response Time (seconds): 1.007
Transient Behavior: 0.74

Step Plot: Reference tracking

System: Tuned response
I/O: In(1) to y
Time (seconds): 0.567
Amplitude: 0.674

Controller Parameters: P = 9.24, I = 21.87, D = 0, N = 100

Controller Parameters

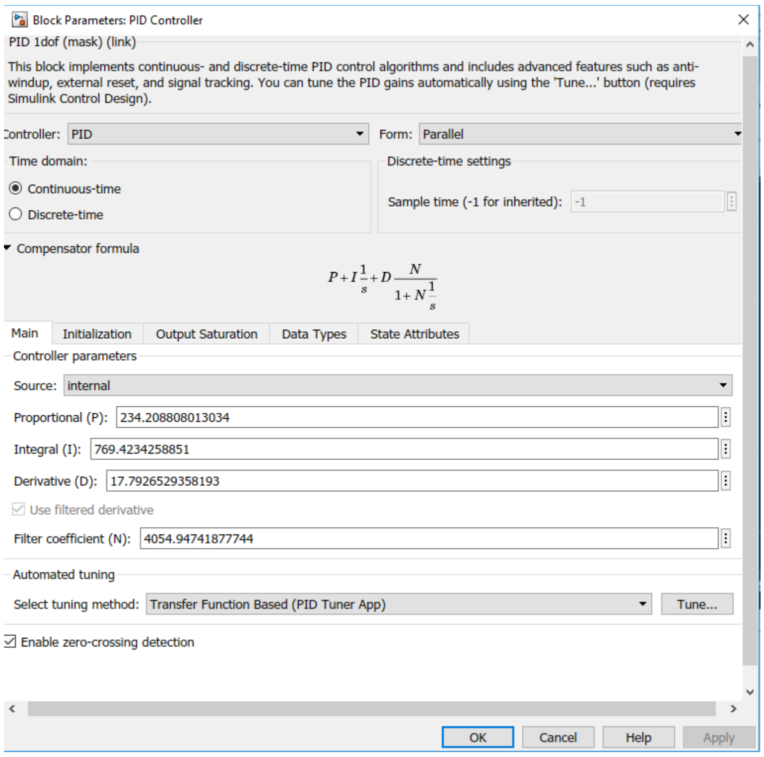
	Tuned	Block
P	9.2396	50
I	21.8691	200
D	0	10
N	100	100

Performance and Robustness

	Tuned	Block
Rise time	0.794 seconds	0.236 seconds
Settling time	1.23 seconds	1.19 seconds
Overshoot	1.26 %	6.81 %
Peak	1.01	1.07
Gain margin	Inf dB @ Inf rad/s	Inf dB @ Inf rad/s
Phase margin	74 deg @ 1.99 rad/s	99.6 deg @ 17.5 rad/s
Closed-loop stability	Stable	Stable

Figure 15,PID tuning window with controls

We can see in Figure 15 that the tuning interface allows us to dynamically modify the Response time and transient behaviour with respect to the settling time and the overshoot. This allows us to achieve a high level of customisation within the controller. below, Figure 16, shows the finalised settings corresponding to the PID controller as implemented.



Block Parameters: PID Controller

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PID
Form: Parallel

Time domain:
[✓ Continuous-time]
Discrete-time

Discrete-time settings
Sample time (-1 for inherited): -1

Compensator formula

$$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$$

Controller parameters
Source: internal
Proportional (P): 234.208808013034
Integral (I): 769.4234258851
Derivative (D): 17.7926529358193
Use filtered derivative ✓
Filter coefficient (N): 4054.94741877744

Automated tuning
Select tuning method: Transfer Function Based (PID Tuner App)

✓ Enable zero-crossing detection

Figure 16, PID controller configuration

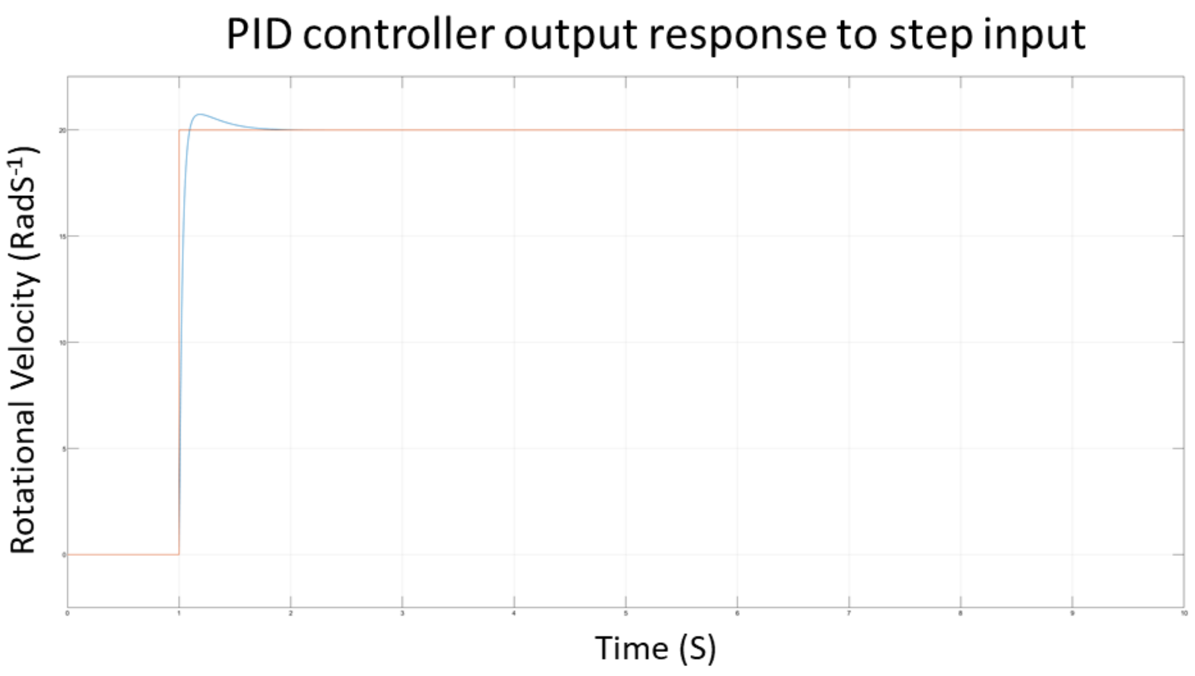


Figure 17, Tuned PID controller Setting time < 2 s, Overshoot < 5%, Steady-state error < 1%.

Figure 17 shows the overall system when we have tuned the PID controller. We can observe the peak of the waveform is at 20.74 RadS^{-1}. Once the time hits 2 seconds, the signal no longer moves outside of 0.01 from the ideal 20RadS^{-1}. This fits firmly within the system specification.

Part 4. Further Discussion

We can see below in Figure 18, the system response to a change in input of a negative step. Whilst this the system is started with the system in equilibrium. The Settings have been kept identical to when the model was tuned in the previous step. We can observe that the system behaves as expected and responds in a similar way proportionally to the input step.

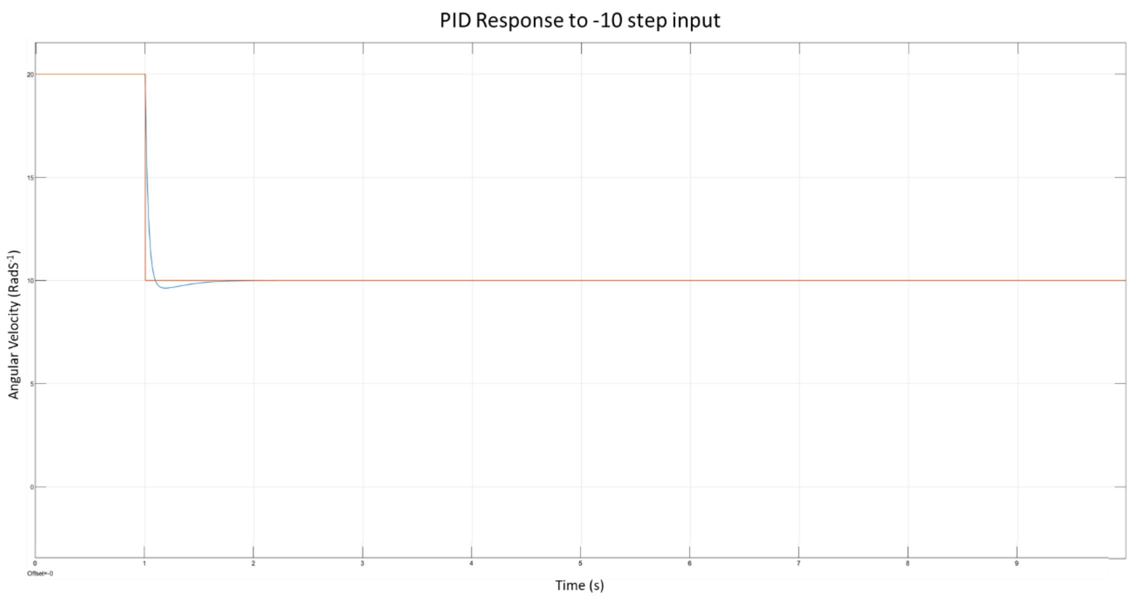


Figure 18, PID change in step response from radial velocity from 20 to 10 RadS^{-1}