

University of Wales Trinity Saint David

Wales Institute of Science & Art



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Modelling Automotive Control Systems in
MATLAB**

Freya Davis-Nunnick

2204780

Motorcycle Engineering MSc

Date (10/04/2026)

Energy Storage and Control Systems ACHA7007

Word count: 1644/1500± 10%

Abstract

This study assessed the sensitivity of a Proportional-Integral-Derivative (PID) controller when the delay was 0.02s and 0.04s. A base MATLAB code was provided to the author. The literature review contains the research assembled for this project this included background reading on controllers, tuning methods and collation of the relevant (ISO). The P, I and D value for the 0.02s delay study were 1.5, 3.5 and 0.06 respectively. The P, I and D value for the 0.04s delay study were 1.7, 2.6 and 0. The controller was more stable, responsive in addition to lesser overshoot and faster settling time with a delay of 0.02 s, compared to 0.04s. Furthermore, a critical analysis of the methods used, and the base code shortcomings were discussed. Lastly, further work to improve the PID controller included ensuring the system meets the relevant ISO standards. In addition, to implementing an automated tuning code.

Table of Contents

1.0	Introduction	8
2.0	Control Theory and Background Information	9
2.1	Controllers Overview	9
2.2	PID Controllers and Tunning Methods	11
2.2.1	Trial and Error Method	11
2.2.2	Ziegler-Nichols Method	11
2.2.3	Cohen-Coon Method	11
2.2.4	Relay Auto-Tunning Method.....	12
2.2.5	Modern Optimisation-Based Methods	12
2.2.6	Comparison	12
2.3	Key equations.....	13
2.4	International Organization for Standardization	15
3.0	MATLAB Implementation	16
3.1	Code Overview	16
3.2	Delay	16
3.3	Encoding and Decoding.....	16
3.4	Tuning	16
4.0	Results	18
4.1	Results for 0.02s Delay.....	18
4.1.1	P Tunning Results.....	18
4.1.2	I Tunning Results When was $P=1.5$	22
4.1.3	D Tunning Results When was $P=1.5$ and $I=3.5$	26
4.1.4	Overview Results When $P=1.5$, $I=3.5$ and $D=0.06$	28
4.1.5	Bit-depth Variation Results.....	29
4.1.6	Sampling Time Variation Results.....	30
4.2	Results for 0.04s Delay.....	31
4.2.1	P Tunning Results.....	31

4.2.2	I Tunning Results When was $P=1.7$	34
4.2.3	D Tunning Results When was $P=1.5$ and $I=2.6$	37
4.2.4	Overview Results When $P=1.7$, $I=2.6$ and $D=0$	40
4.2.5	Bit-depth Variation Results	41
4.2.6	Sampling Time Variation Results	42
5.0	Analysis	43
5.1	Analysis for 0.02s Delay	43
5.1.1	P Tunning Analysis	43
5.1.2	I Tunning Analysis	44
5.1.3	D Tunning Analysis	45
5.1.4	Overview Analysis	46
5.1.5	Bit-depth Variation Analysis	47
5.1.6	Sampling Time Variation Analysis	48
5.2	Analysis for 0.04s Delay	50
5.2.1	P Tunning Analysis	50
5.2.2	I Tunning Analysis	51
5.2.3	D Tunning Analysis	53
5.2.4	Overview Analysis	53
5.2.5	Bit-depth Variation Analysis	54
5.2.6	Sampling Time Variation Analysis	55
6.0	Discussion & Conclusions	57
6.1	Model Weaknesses	57
6.2	Conclusions	57
7.0	Future work	58
7.1	Automated Tunning	58
7.2	Acceleration Limits	58
8.0	References	59

Table of Figures

Figure 1: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.02s, 70s of Simulation (full)	18
Figure 2: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.02s, 10s of Simulation	19
Figure 3: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.02s, 2s of Simulation	19
Figure 4: Graph for Further P Value Tunning when I and D are 0 and Delay is 0.02s, 70s of Simulation (full)	20
Figure 5: Graph for Further P Value Tunning when I and D are 0 and Delay is 0.02s, 10s of Simulation	20
Figure 6: Graph for Further P Value Tunning when I and D are 0 and Delay is 0.02s, 2s of Simulation	21
Figure 7: Graph for Initial I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 70s of Simulation (full)	22
Figure 8: Graph for Initial I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 10s of Simulation	23
Figure 9: Graph for Initial I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 2s of Simulation	23
Figure 10: Graph for Further I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 70s of Simulation (full)	24
Figure 11: Graph for Further I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 10s of Simulation	24
Figure 12: Graph for Further I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 2s of Simulation	25
Figure 13: Graph for Further I Value Tunning when P=1.5, D=0 and Delay is 0.02s, 35s to 40s of Simulation Close Up	25
Figure 14: Graph for D Value Tunning when P=1.5, I=3.5 and Delay is 0.02s, 70s of Simulation (full)	26
Figure 15: Graph for D Value Tunning when P=1.5, I=3.5 and Delay is 0.02s, 10s of Simulation).....	27
Figure 16: Graph for D Value Tunning when P=1.5, I=3.5 and Delay is 0.02s, 2s of Simulation	27
Figure 17: Graph for Overview when P=1.5, I=3.5, D=0.06 and Delay is 0.02s, 70s of Simulation (full)	28

Figure 18: Graph for Bit-Depth Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation	29
Figure 19: Graph for Sampling Time Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation	30
Figure 20: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)	31
Figure 21: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.04s, 10s of Simulation	31
Figure 22: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.04s, 2s of Simulation	32
Figure 23: Graph for Further P Value Tunning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)	32
Figure 24: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.04s, 10s of Simulation	33
Figure 25: Graph for Initial P Value Tunning when I and D are 0 and Delay is 0.04s, 2s of Simulation (full)	33
Figure 26: Graph for Initial I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)	34
Figure 27: Graph for Initial I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 10s of Simulation	34
Figure 28: Graph for Initial I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 2s of Simulation	35
Figure 29: Graph for Further I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)	35
Figure 30: Graph for Further I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 10s of Simulation	36
Figure 31: Graph for Further I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 2s of Simulation	36
Figure 32: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 70s of Simulation (full)	37
Figure 33: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 10s of Simulation	37
Figure 34: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 2s of Simulation	38
Figure 35: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 34.5s to 37.5s of Simulation Close Up	38

Figure 36: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 25s to 28s of Simulation Close Up	39
Figure 37: Graph for Overview when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 70s of Simulation (full).....	40
Figure 38: Graph for Bit-Depth Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 3s of Simulation	41
Figure 39: Graph for Sampling Time Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 3s of Simulation	42
Figure 40: Graph of Acceleration Limits from ISO	58

Table of Tables

Table 1: Definitions for PID Controllers and Tunning	9
Table 2: Table of Controller Types	10
Table 3: Table of Relevant ISO Standards.....	15
Table 4: P, I and D Values for the Two Delay Studies	43

Table of Equations

Equation 1: Output of a Proportional Controller	13
Equation 2: Error Signal Value	13
Equation 3: Output of a Proportional-Integral Controller	13
Equation 4: Output of a Proportional-Integral-Derivative Controller	14
Equation 5: Bits Resolution	14
Equation 6: Discrete Derivative Approximation.....	14
Equation 7: Discrete Integral Approximation.....	14

1.0 Introduction

Controllers are used to manage systems such as HVAC systems, chemical refining, power generation. This report pertains to the use of them in the automotive sector and namely focuses on the use of Proportional–Integral–Derivative (PID) controllers. A PID controller was emulated using MATLAB and was then tuned to improve accuracy. This study aims to complete a sensitive study of the PID controller in MATLAB by investigating the effects of delay, sampling time, and bit resolution.

2.0 Control Theory and Background Information

2.1 Controllers Overview

Control systems are used to regulate dynamic systems. A controller continuously compares the system output with its demanded value and applies action to minimise the error. One of the most widely used control strategies is the Proportional–Integral–Derivative (PID) controller. Table 1 summarises and defines the key components of a PID controller and their influence on tuning. (Tandan and Kumar Swarnkar, 2015; Bennett *et al.*, 2023)

Table 1: Definitions for PID Controllers and Tuning

Term	Definition	Effects on Tuning
P	Produces an output that is proportional to the error signal	Affects response time Greater P values decrease response time but can cause overshoot
I	Accumulates previous error signal values and integrates them over time	Used to eliminate steady-state error Greater I values can cause overshoot (integral windup)
D	Accounts for the rate of change of the error signal	Used to reduce overshoot, settling time and improve stability
Delay	The time difference between control signal output and system response	Greater delays decrease ease of tuning of the system and reduce the stability of the system
Overshoot	The magnitude the system exceeds the demand by	Often due to unnecessarily large P and I values Increasing D reduces overshoot
Settling Time	Time taken for the system to reach and remain within a small range of the target value e.g. when the systems output is within 2% of demand	Greater P values reduce settling time Correct D value assists the system in settling faster

Steady-State Error	The remaining error once the system has settled	Increase P reduces the steady-state error but cannot eliminate it completely. Increasing I further reduces the steady-state error
--------------------	---	--

There are many different controllers, they can all be grouped into two types: open-loop and closed-loop. Open-loop controllers have no feedback. Therefore, the input and the output have no communication. These controllers are simple but can be inaccurate as the controller cannot self-assess and correct for errors. Closed-loop have a feedback loop and therefore can self-correct. These are more complex systems than open-loop controllers but are capable of a more accurate output. An example of a closed-loop controller would be a climate control system; the temperature is monitored and if it no longer aligns with the demanded the system is activated. Table 2 provides a comparison of commonly used controller types, including their applications, advantages, and limitations

Table 2: Table of Controller Types

Controller	Common Uses	Advantages	Disadvantages
P	Basic speed control and simple temperature control	Simple implementation, fast response, low computational cost	Steady-state error present, high gain can cause oscillation or instability
PI	Cruise control, voltage regulation and throttle control	Eliminates steady-state error, good disturbance rejection, relatively simple	Slower response than P, risk of integral wind-up, increased overshoot
PID	Cruise control, engine management and ABS	Fast response, eliminates steady-state error, improved damping and stability	Sensitive to noise (derivative term), tuning complexity, higher computational cost
PD	Position control and suspension systems	Improves stability and damping, reduces overshoot	Does not eliminate steady-state error, sensitive to measurement noise

2.2 PID Controllers and Tuning Methods

There are several methods for tuning PID controllers; a few are discussed in this chapter. Trial and error tuning is commonly used due to its simplicity. Classical methods such as the Ziegler–Nichols and Cohen–Coon techniques provide systematic ways of determining controller parameters. More advanced approaches, including Internal Model Control (IMC) and relay auto-tuning, use system models or automated procedures to improve performance and stability. Modern optimisation-based methods further elevate tuning accuracy.

2.2.1 Trial and Error Method

This method involves adjusting the gains iteratively based on system response. Typically, the proportional gain is increased first, followed by integral and derivative adjustments. (Bennett *et al.*, 2023). The equations in chapter 2.3 can aid with manual tuning.

2.2.2 Ziegler-Nichols Method

It involves increasing the proportional gain until the system reaches sustained oscillations (ultimate gain), then uses the oscillation period to calculate PID parameters. (Patel, 2020) (Sung, Lee and Lee, 1998)

2.2.3 Cohen-Coon Method

This method uses the process reaction curve (step response) to determine PID parameters. The ultimate gain and frequency can be found using the Cohen-Coon method and then the 'Ziegler–Nichols tuning rule can be used'. (Sung, Lee and Lee, 1998)

2.2.4 Relay Auto-Tuning Method

The relay auto-tuning method introduces a relay to force oscillations in the system. The resulting oscillation data is used to calculate PID gains automatically. (Sung, Lee and Lee, 1998)

2.2.5 Modern Optimisation-Based Methods

These methods use computational algorithms (e.g. machine learning) to automatically tune PID parameters based on performance criteria. (Om Bansal, Sharma and Shreeraman, 2012)

2.2.6 Comparison

Methods such as Trial and error and Ziegler–Nichols are simple and widely used however, they often require refinement after initial tuning. Whereas modern optimisation-based methods provide improved performance at the cost of increased computational complexity.

2.3 Key equations

$$u(t) = K_p \times e(t)$$

Equation 1: Output of a Proportional Controller

$u(t)$ =Controller output signal

K_p =Proportional gain

$e(t)$ =Error signal

$$e(t) = r(t) - y(t)$$

Equation 2:Error Signal Value

$e(t)$ =Error signal

$r(t)$ =Target value

$y(t)$ =True value

$$u(t) = K_p \times e(t) + K_i \int e(t) dt$$

Equation 3: Output of a Proportional-Integral Controller

$u(t)$ =Controller output signal

K_p =Proportional gain

$e(t)$ =Error signal

K_i =Integral gain

$\int e(t) dt$ =Integral of error over time

$$u(t) = K_p \times e(t) + K_i \int e(t) + K_d \times \frac{de(t)}{dt}$$

Equation 4: Output of a Proportional-Integral-Derivative Controller

$u(t)$ =Controller output signal

K_p =Proportional gain

$e(t)$ =Error signal

K_i =Integral gain

$\int e(t)dt$ =Integral of error over time

K_d =Derivative gain

$\frac{de(t)}{dt}$ =Rate of change of error

$$\Delta = \frac{X_{max} - X_{min}}{2^n}$$

Equation 5: Bits Resolution

Δ =Resolution

$X_{max} - X_{min}$ =Signal range

n =Number of bits

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T_s}$$

Equation 6: Discrete Derivative Approximation

$e(k)$ =Sampling time

T_s =Discrete time index

$$\int e(t)dt \approx \Sigma e(k) \times T_s$$

Equation 7: Discrete Integral Approximation

$e(t)$ =Error signal

$e(k)$ =Sampling time

T_s =Discrete time index (time step)

2.4 International Organization for Standardization

The international organization for standardization have numerous standards (ISO) that cover the behaviour of cruise control systems. They exist to improve the safety of these systems.

Table 3: Table of Relevant ISO Standards

Standard	Description	Relevance
ISO 15622	'Basic control strategy, minimum functionality requirements, basic driver interface elements, minimum requirements for diagnostics and reaction to failure, and performance test procedures for Adaptive Cruise Control (ACC) systems.' (International Organization for Standardization, 2018a)	Speed regulation and system response Acceptable deacceleration of less than or equal to 3.5m/s^2 and rate of change of deacceleration of less than or equal to 2.5m/s^3
ISO 26262	'Safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production road vehicles' (International Organization for Standardization, 2018b)	The vehicle is required to have system to prevent issues from arising due to errors and malfunction from the PID controller
ISO 21448	'Framework and guidance on measures to ensure the safety of the intended functionality (SOTIF), which is the absence of unreasonable risk due to a hazard caused by functional insufficiencies' (International Organization for Standardization, 2022)	Ensures the controller behaviour does not induce risks under unexpected conditions

3.0 MATLAB Implementation

3.1 Code Overview

MATLAB files were supplied; the main script simulated PID controller for a cruise control system in a car, the functions simulated the vehicle and a driver who demanded different speeds via the cruise control script. Errors were identified and corrected and the controller tuned to improve the performance.

3.2 Delay

The delay was set to 0.02s. Typical sampling times used in control systems were fi0.01s–0.05s, with research showing implementations using times of approximately 0.02 seconds. This this improved the realism of the simulation. A further study was conducted where the delay was set to 0.04s. This was completed to assess the tuneability of the controller with a greater delay value.(Halder, Das and Gupta, 2021)

3.3 Encoding and Decoding

The first correction was to insert 'bin2dec' and 'dec2bin' in the appropriate places. The code required discrete digital values (binary) to transmit through the communication bus while other calculations used base-10.

3.4 Tuning

A combination of trial and error and Ziegler-Nichols method was used.

The delay was set to 0.02s.

The greatest P value that did not result in un-tunable overshoot was selected. The I value that minimised the steady-error and settling time without introducing un-tunable overshoot

was selected. The D that to reduce overshoot without amplifying velocity noise was selected. Once the gain values were set a bit-rate study was ran in addition to a sampling time study.

This was then repeated for a delay of 0.04s.

4.0 Results

4.1 Results for 0.02s Delay

4.1.1 P Tuning Results

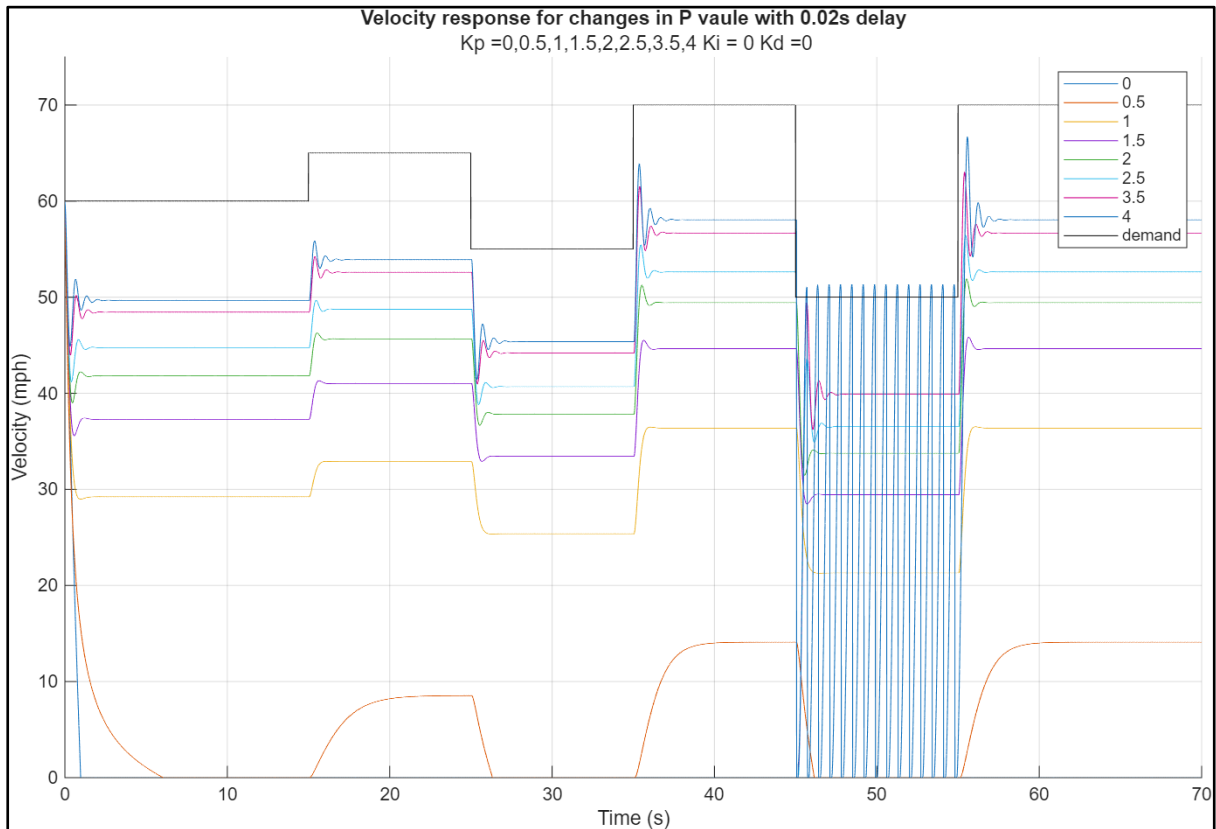


Figure 1: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.02s, 70s of Simulation (full)

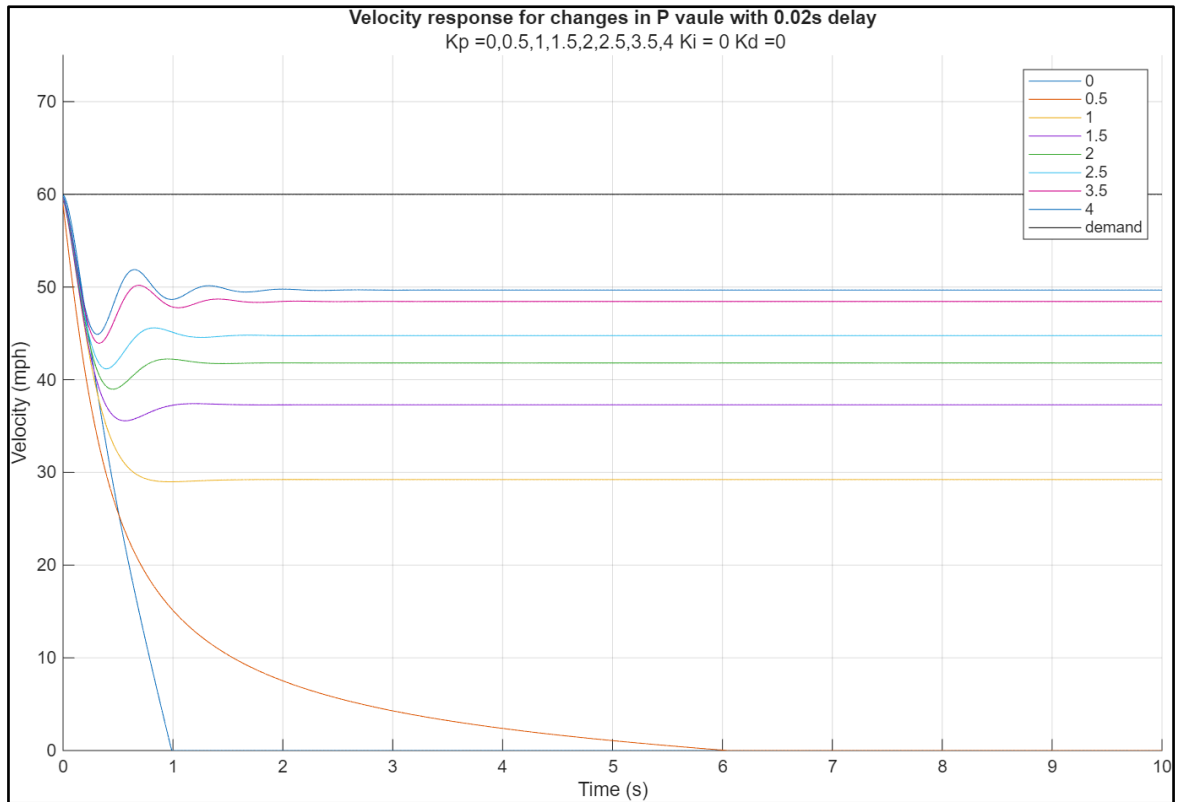


Figure 2: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.02s, 10s of Simulation

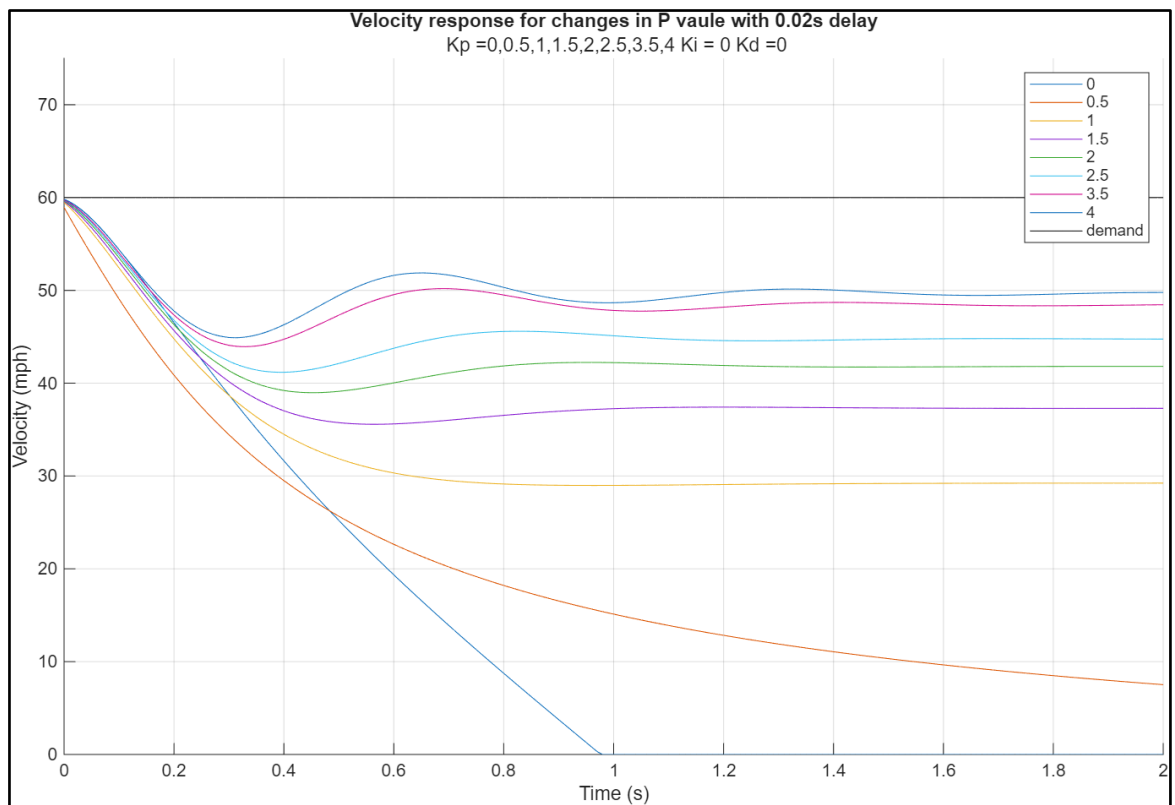


Figure 3: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.02s, 2s of Simulation

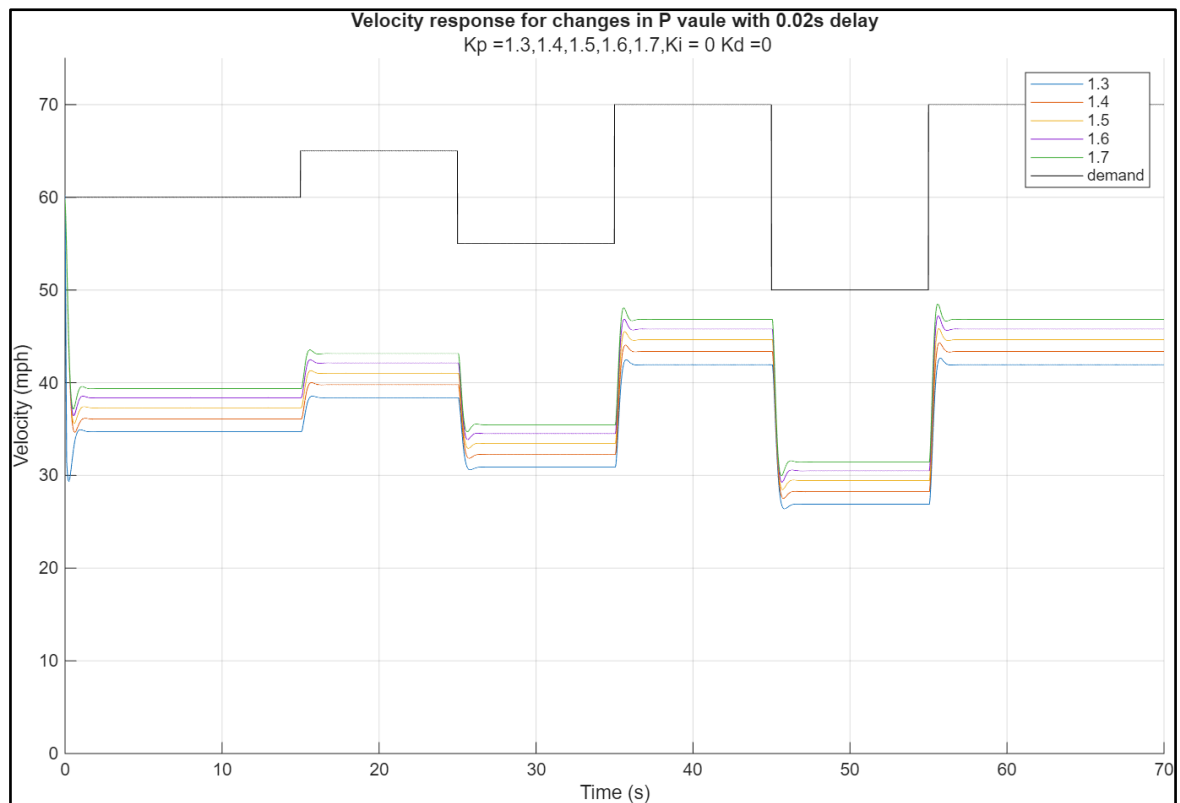


Figure 4: Graph for Further P Value Tuning when I and D are 0 and Delay is 0.02s, 70s of Simulation (full)

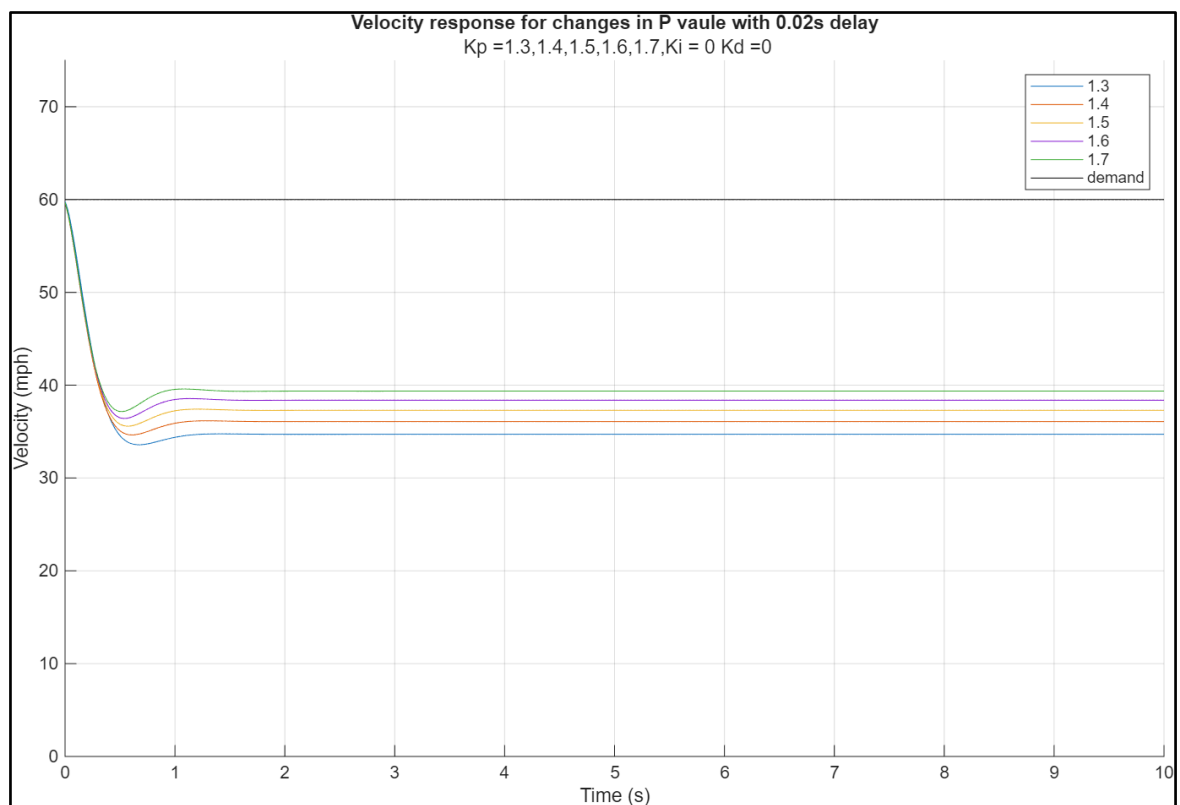


Figure 5: Graph for Further P Value Tuning when I and D are 0 and Delay is 0.02s, 10s of Simulation

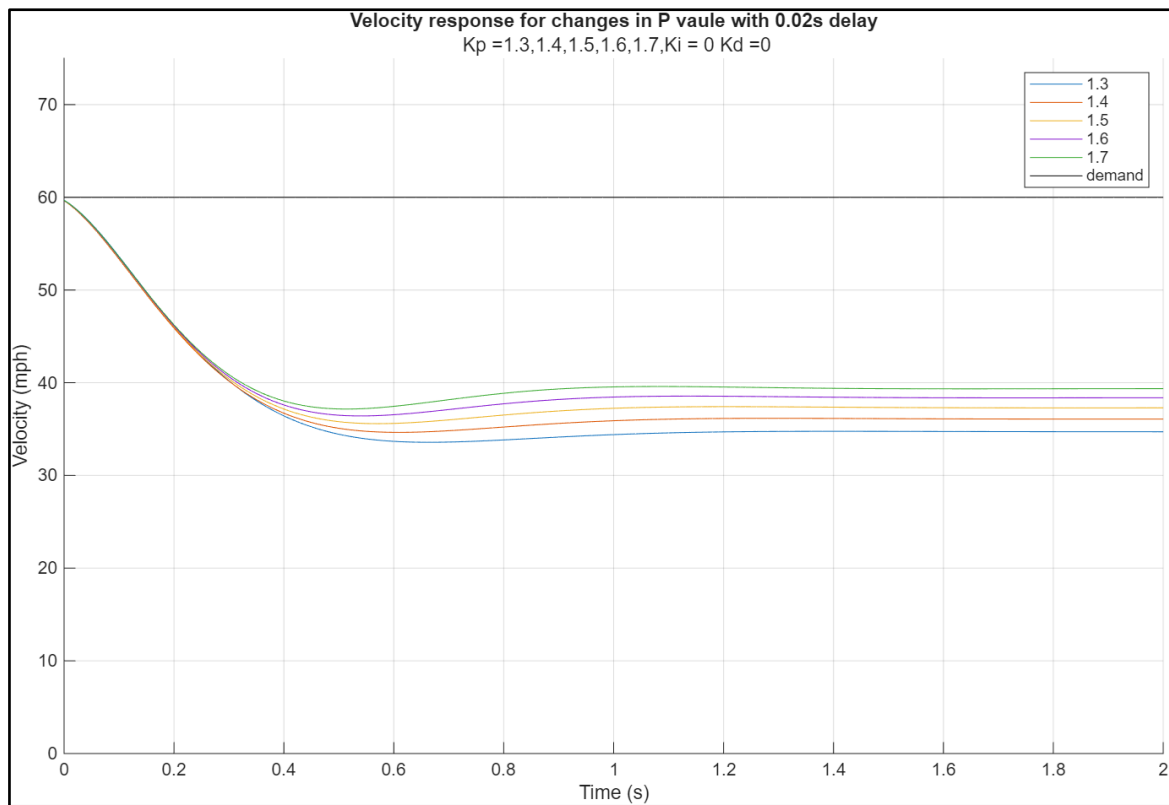


Figure 6: Graph for Further P Value Tunning when I and D are 0 and Delay is 0.02s, 2s of Simulation

4.1.2 I Tunning Results When was $P=1.5$

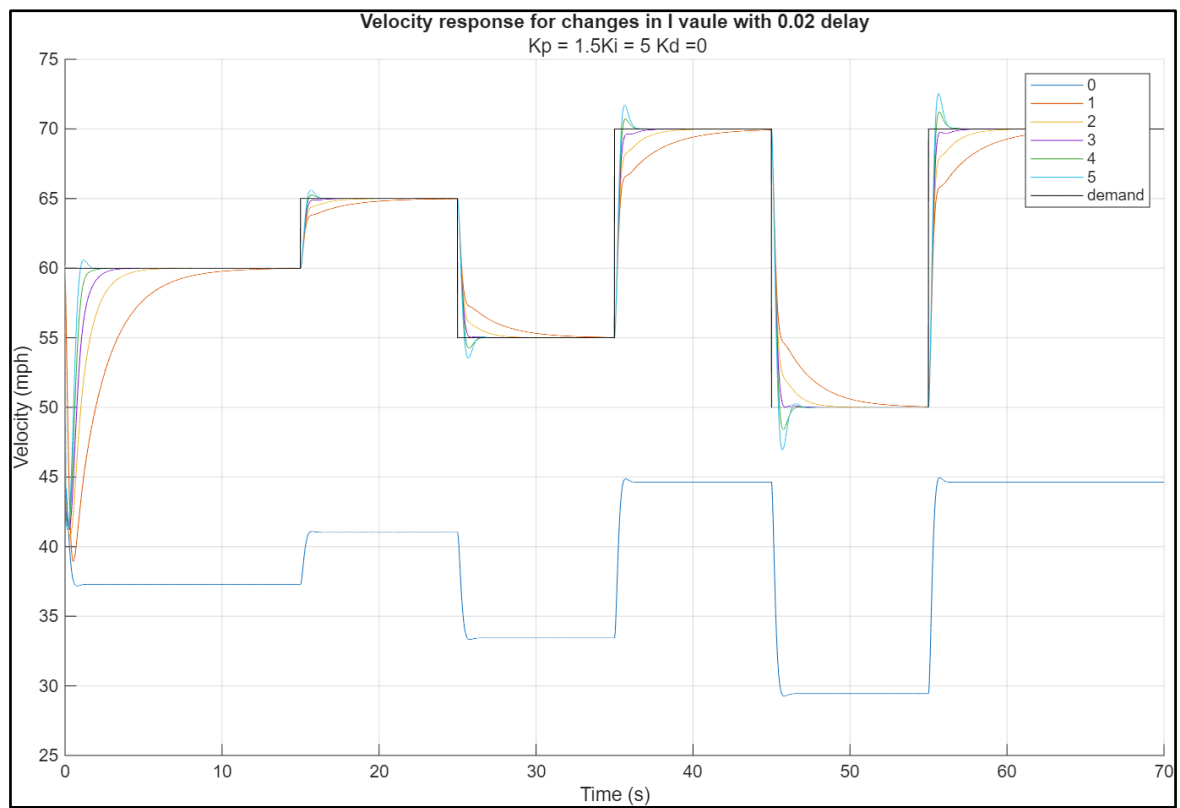


Figure 7: Graph for Initial I Value Tuning when $P=1.5$, $D=0$ and Delay is 0.02s, 70s of Simulation (full)

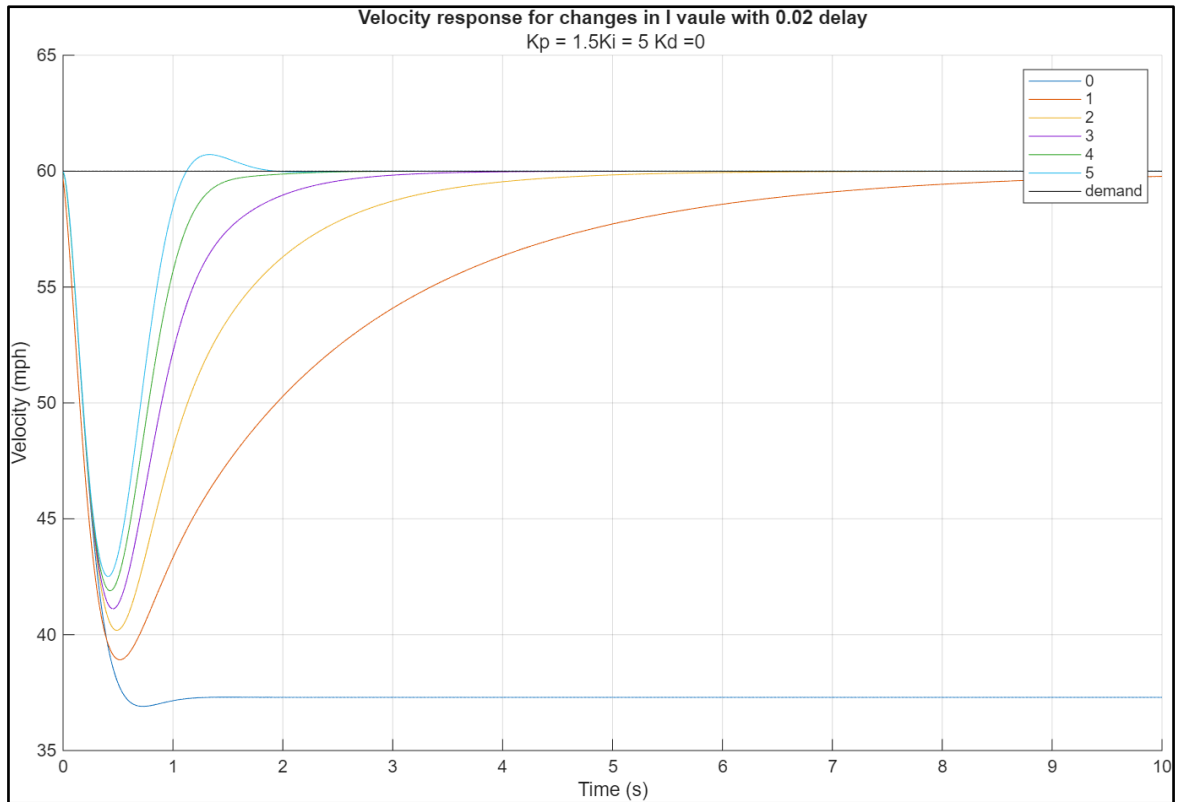


Figure 8: Graph for Initial I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.02s, 10s of Simulation

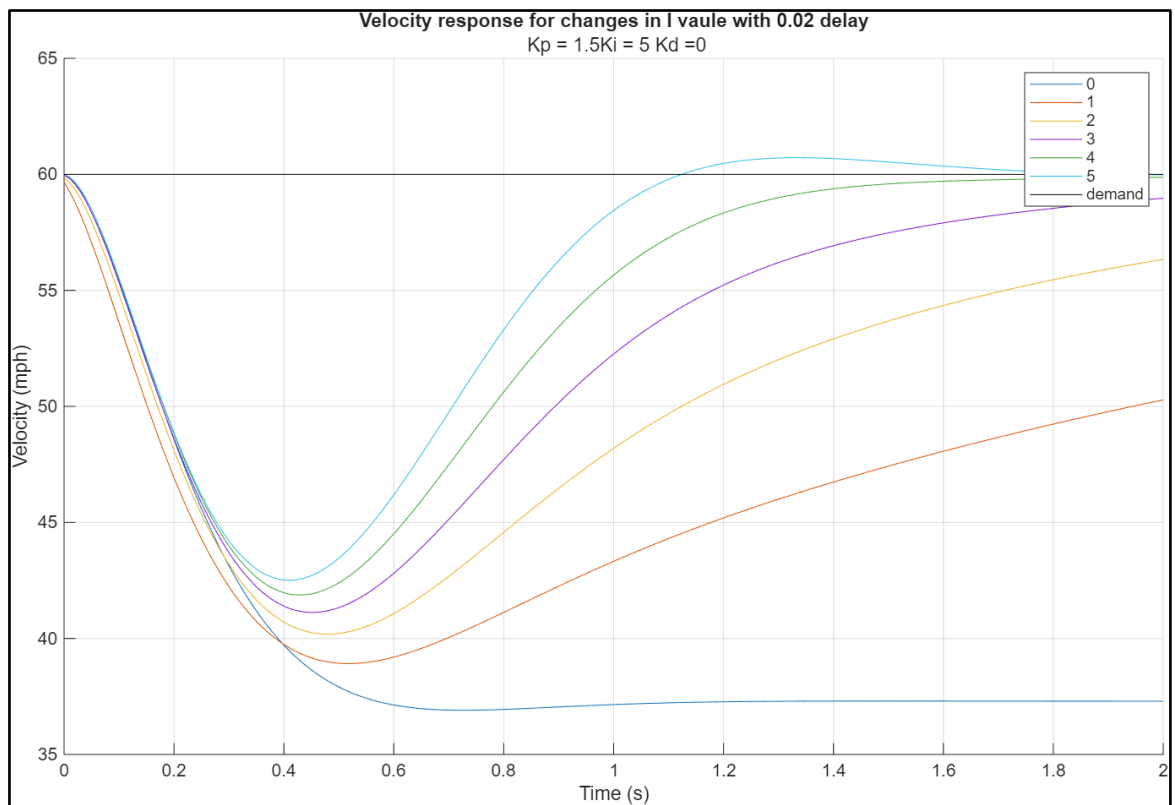


Figure 9: Graph for Initial I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.02s, 2s of Simulation

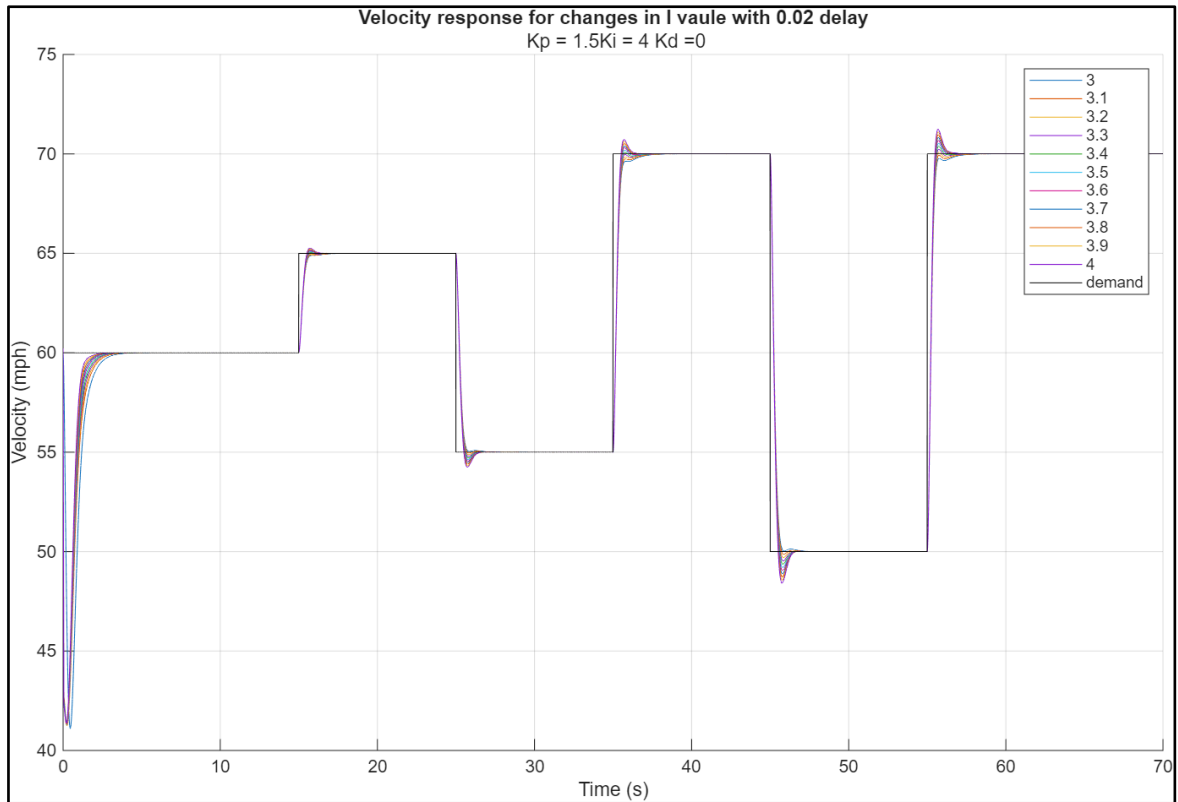


Figure 10: Graph for Further I Value Tuning when $P=1.5$, $D=0$ and Delay is 0.02s, 70s of Simulation (full)

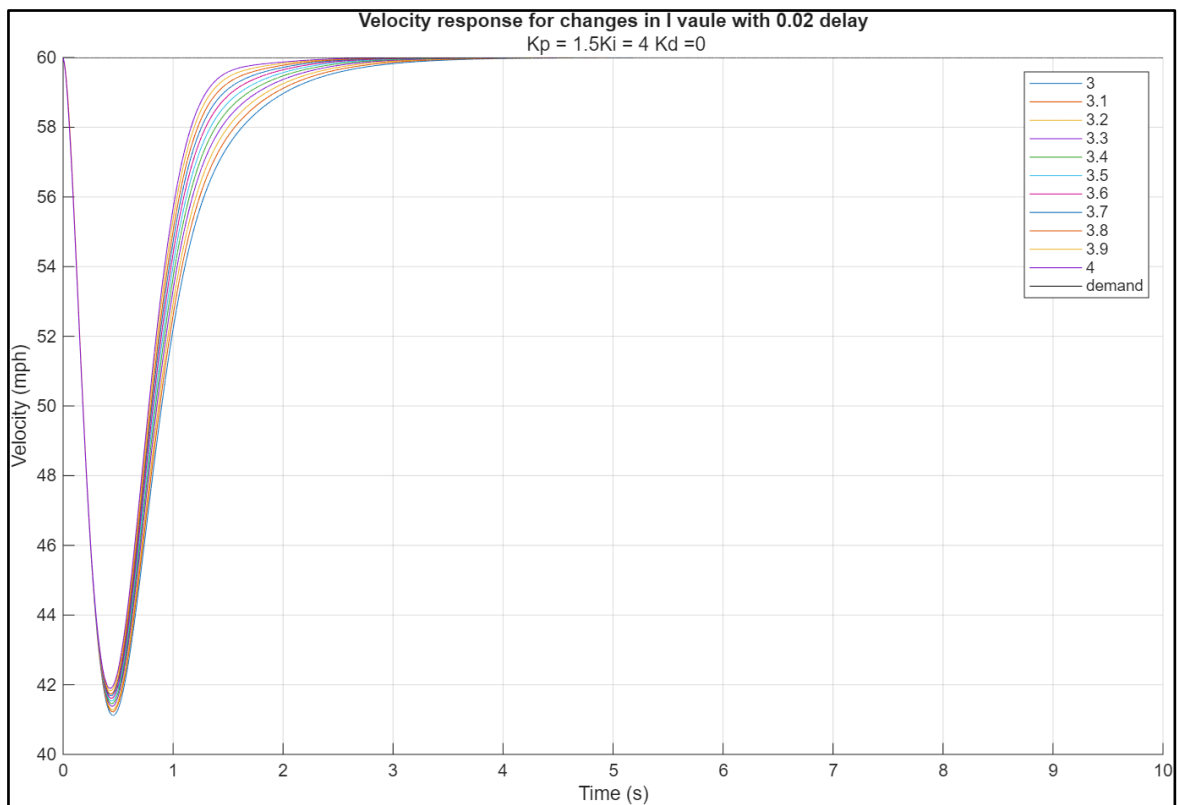


Figure 11: Graph for Further I Value Tuning when $P=1.5$, $D=0$ and Delay is 0.02s, 10s of Simulation

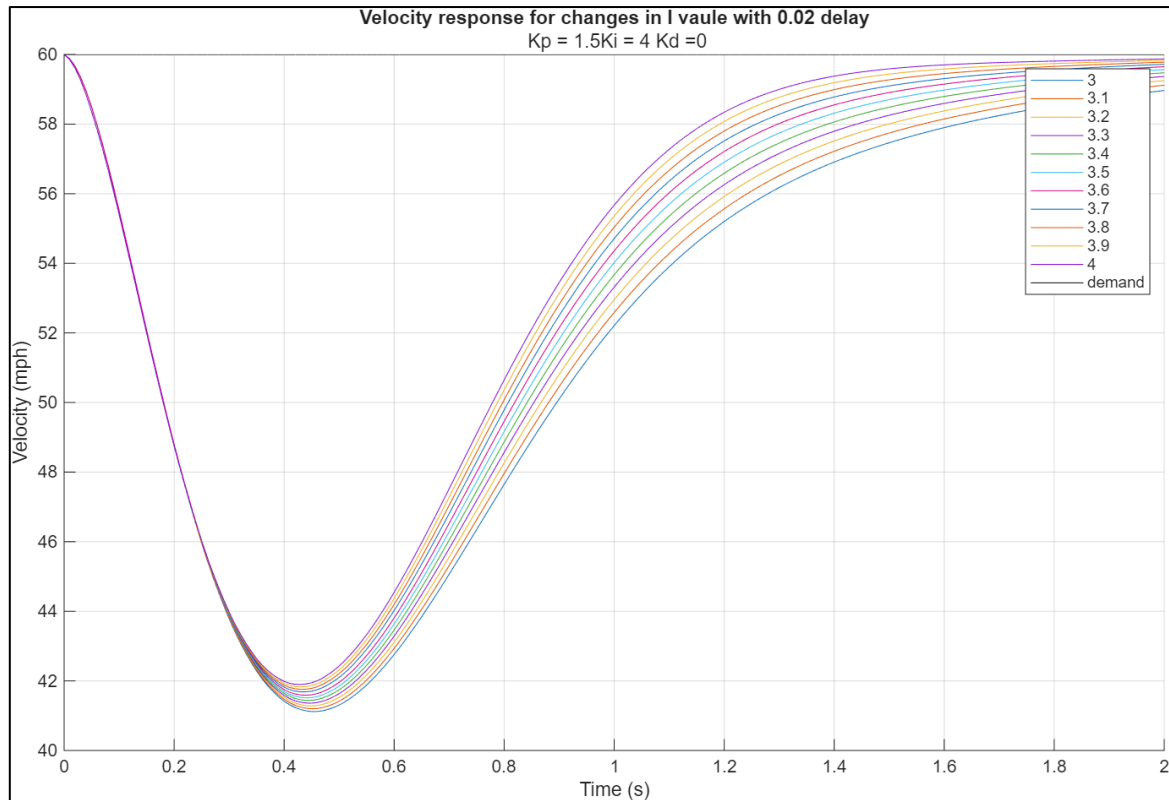


Figure 12: Graph for Further I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.02s, 2s of Simulation

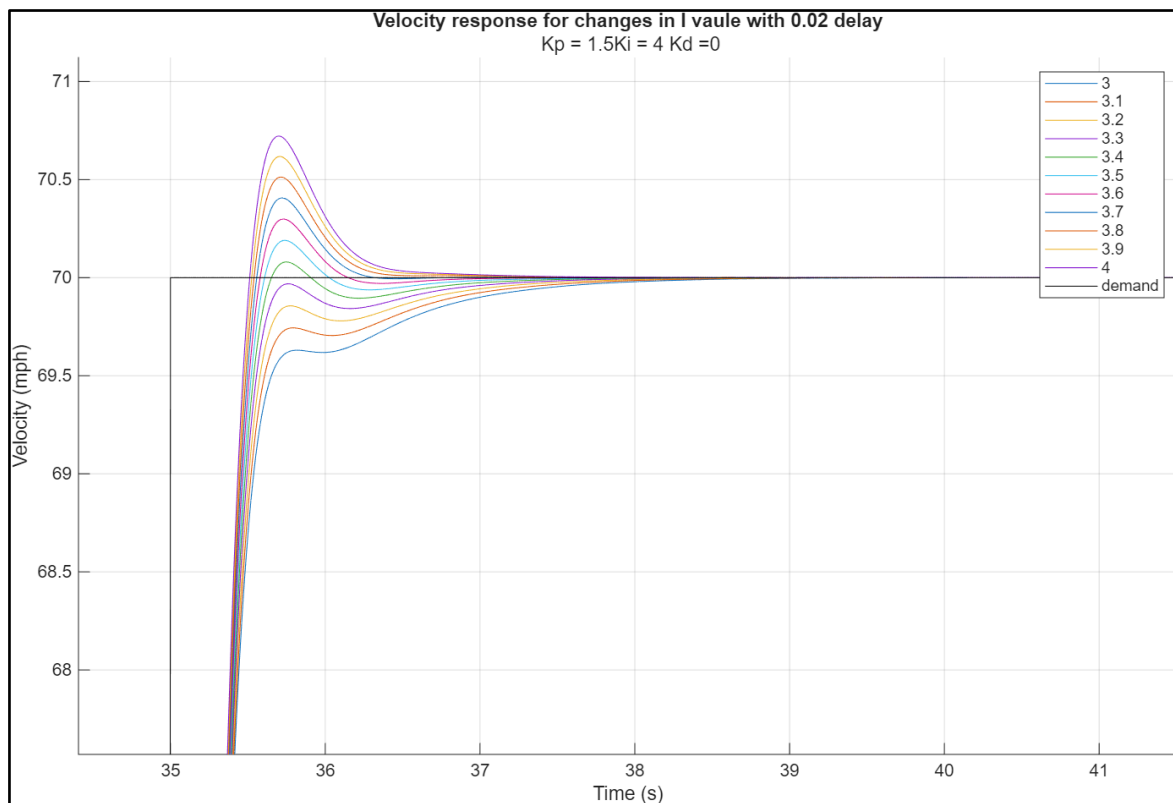


Figure 13: Graph for Further I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.02s, 35s to 40s of Simulation Close Up

4.1.3 D Tunning Results When was $P=1.5$ and $I=3.5$

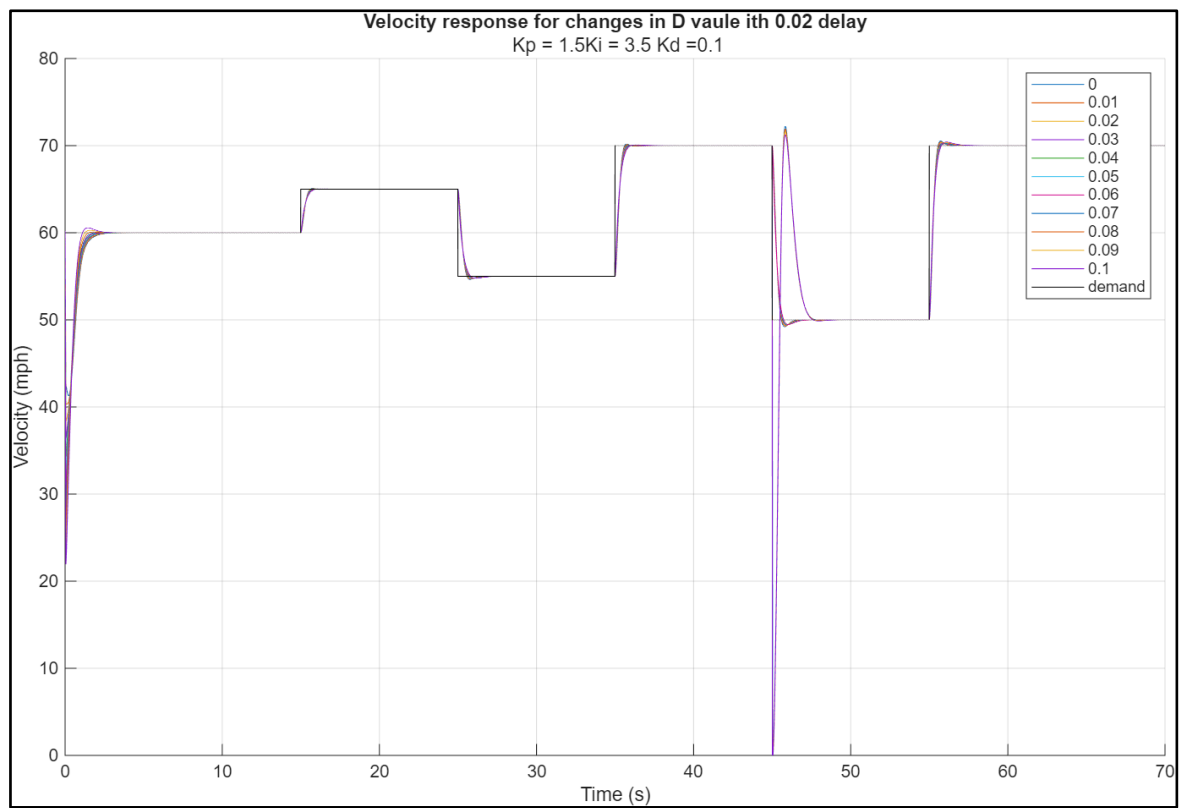


Figure 14: Graph for D Value Tuning when $P=1.5$, $I=3.5$ and Delay is 0.02s, 70s of Simulation (full)

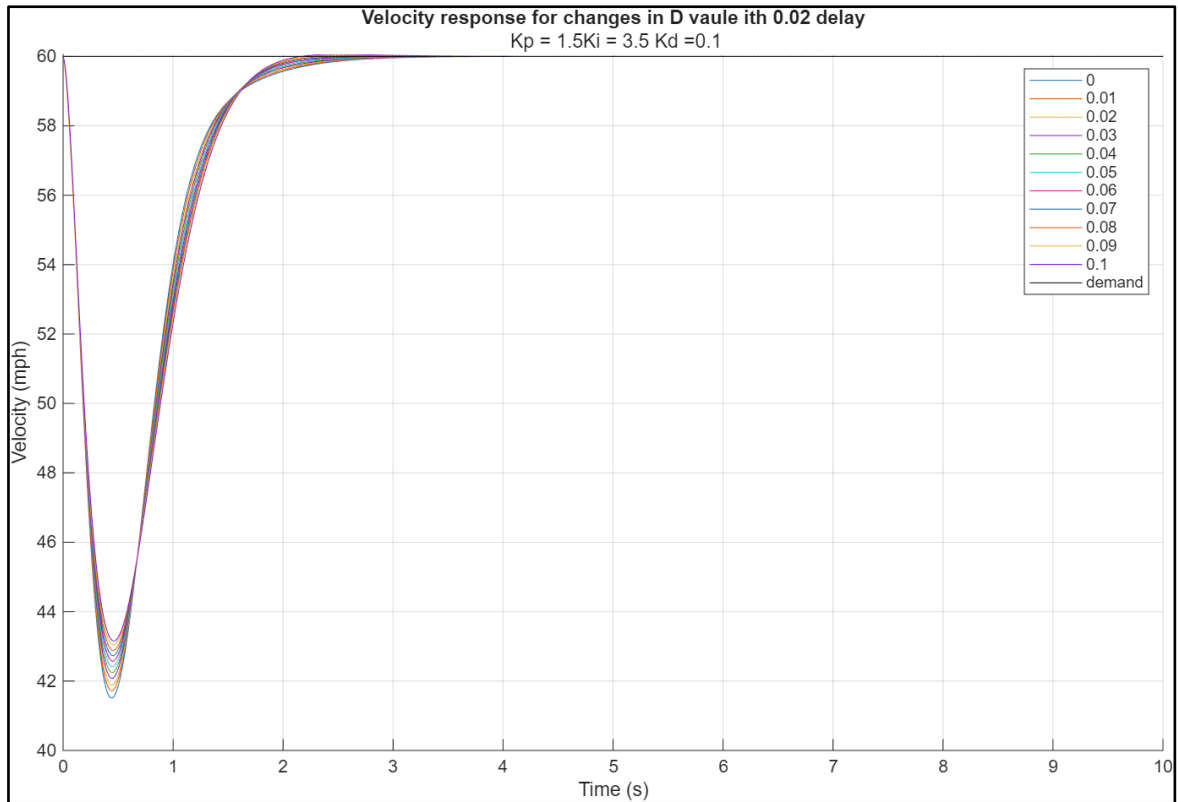


Figure 15: Graph for D Value Tuning when $P=1.5$, $I=3.5$ and Delay is 0.02s, 10s of Simulation)

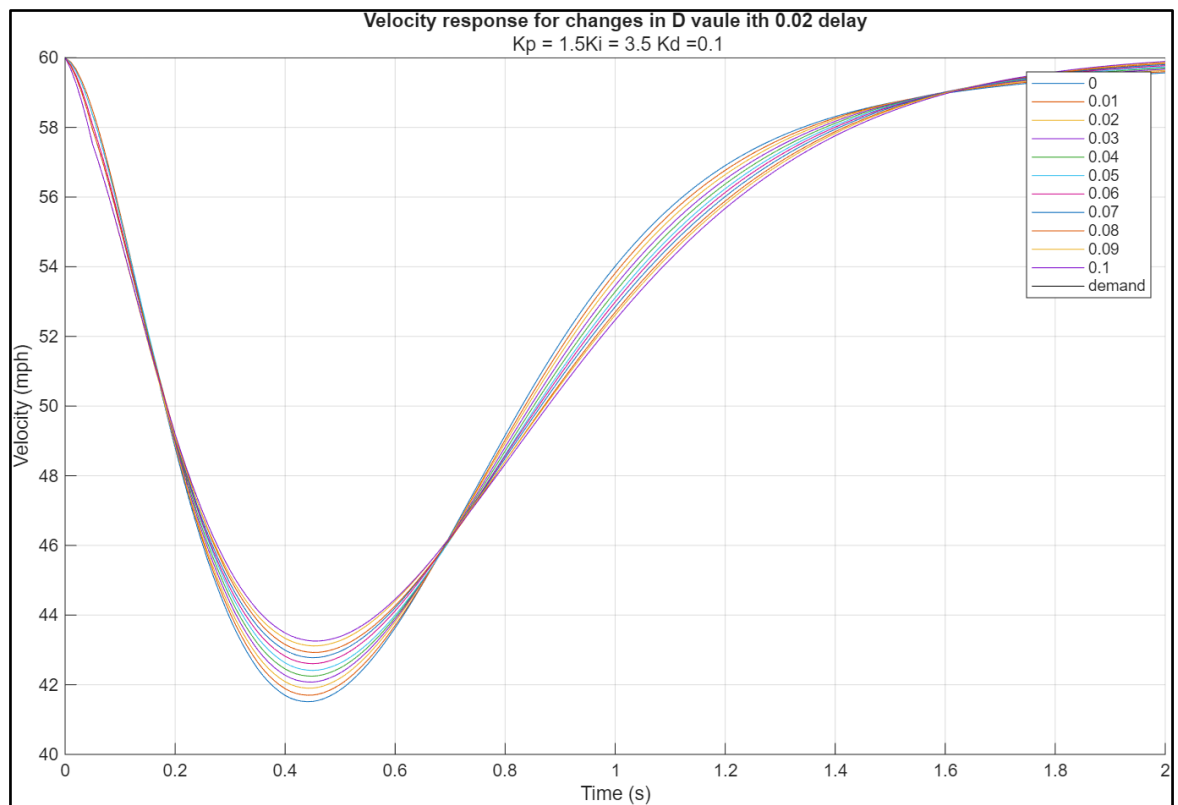


Figure 16: Graph for D Value Tuning when $P=1.5$, $I=3.5$ and Delay is 0.02s, 2s of Simulation

4.1.4 Overview Results When $P=1.5$, $I=3.5$ and $D=0.06$

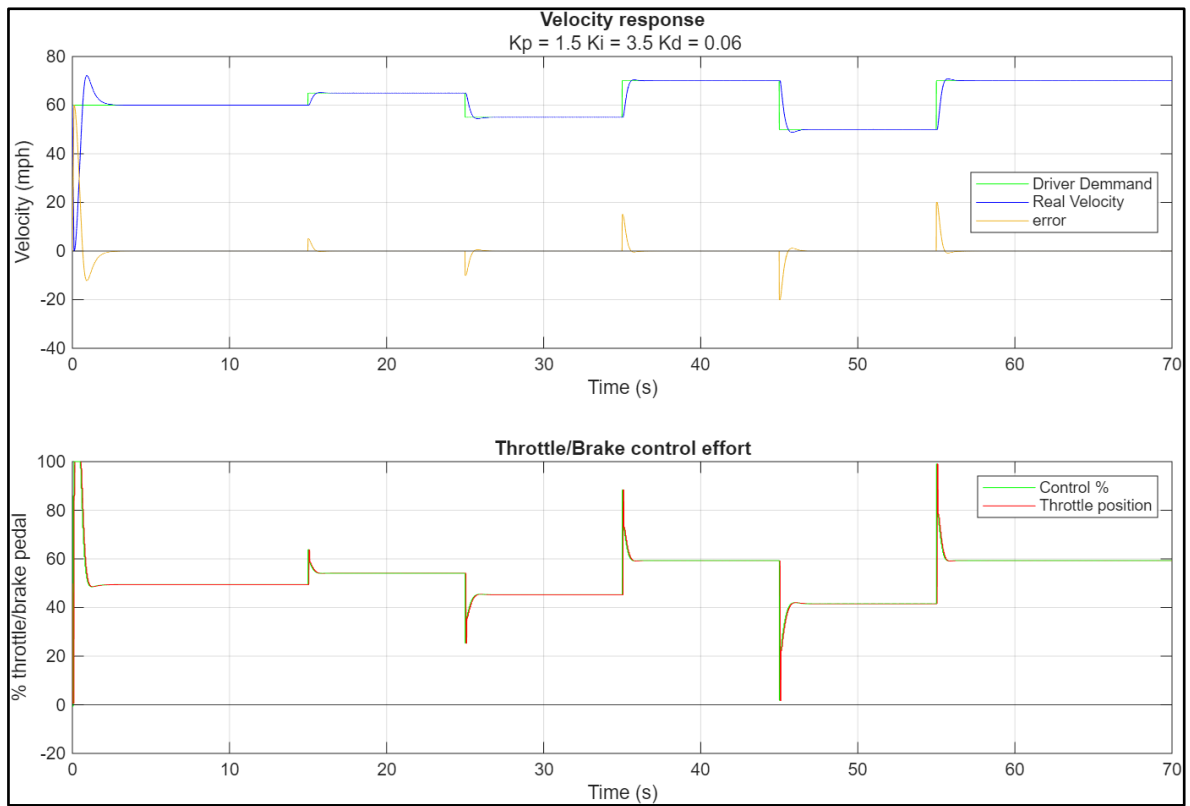


Figure 17: Graph for Overview when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 70s of Simulation (full)

4.1.5 Bit-depth Variation Results

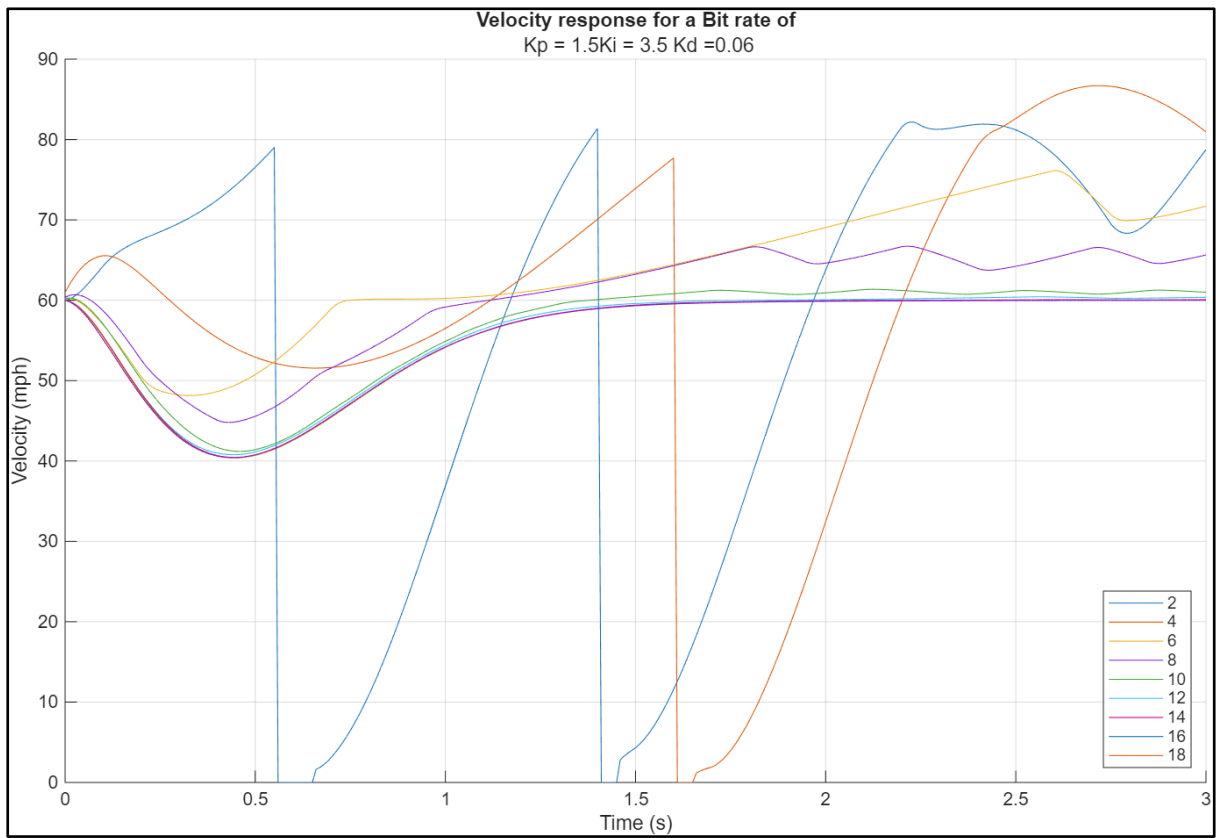


Figure 18: Graph for Bit-Depth Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation

4.1.6 Sampling Time Variation Results

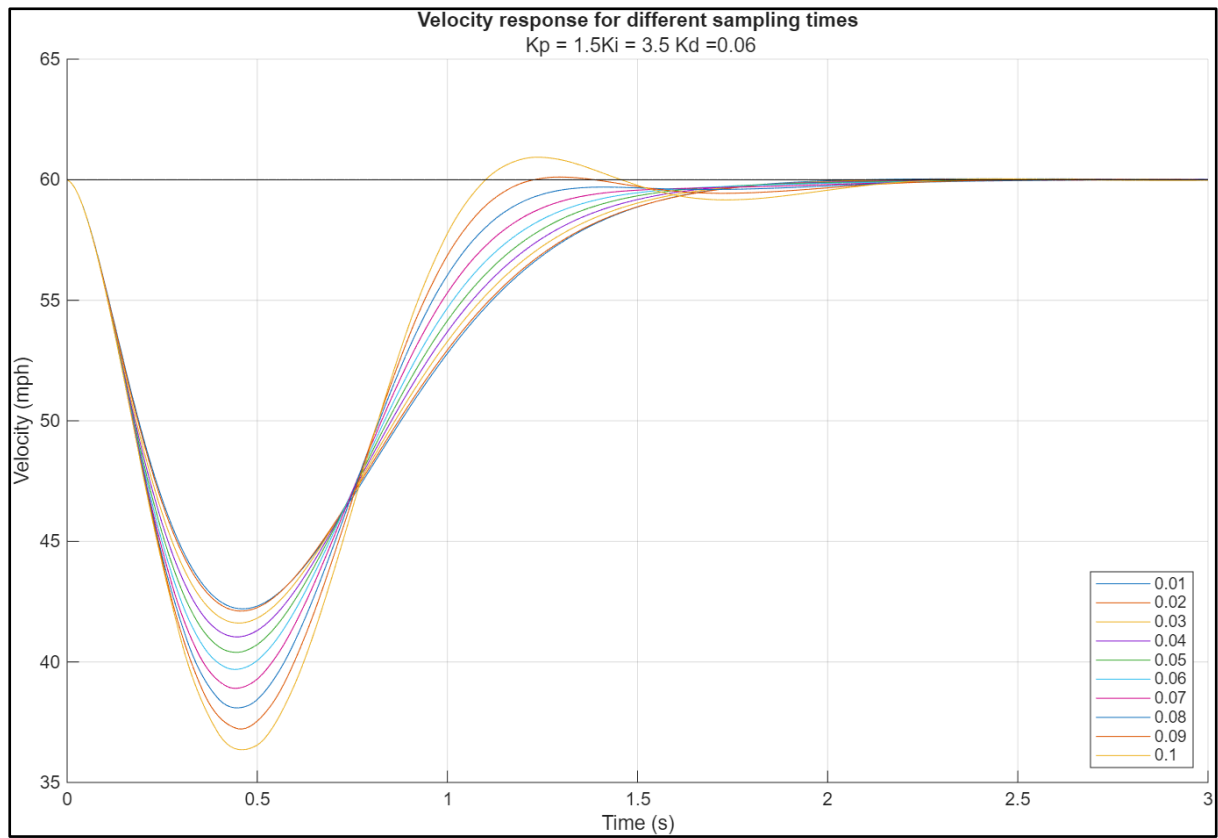


Figure 19: Graph for Sampling Time Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation

4.2 Results for 0.04s Delay

4.2.1 P Tuning Results

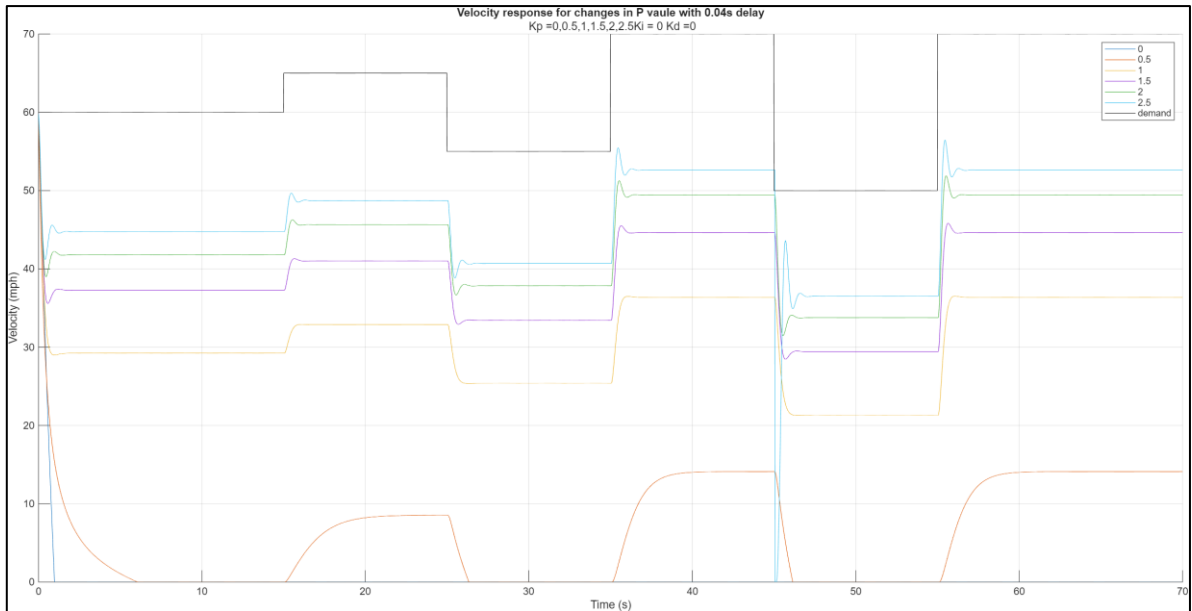


Figure 20: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)

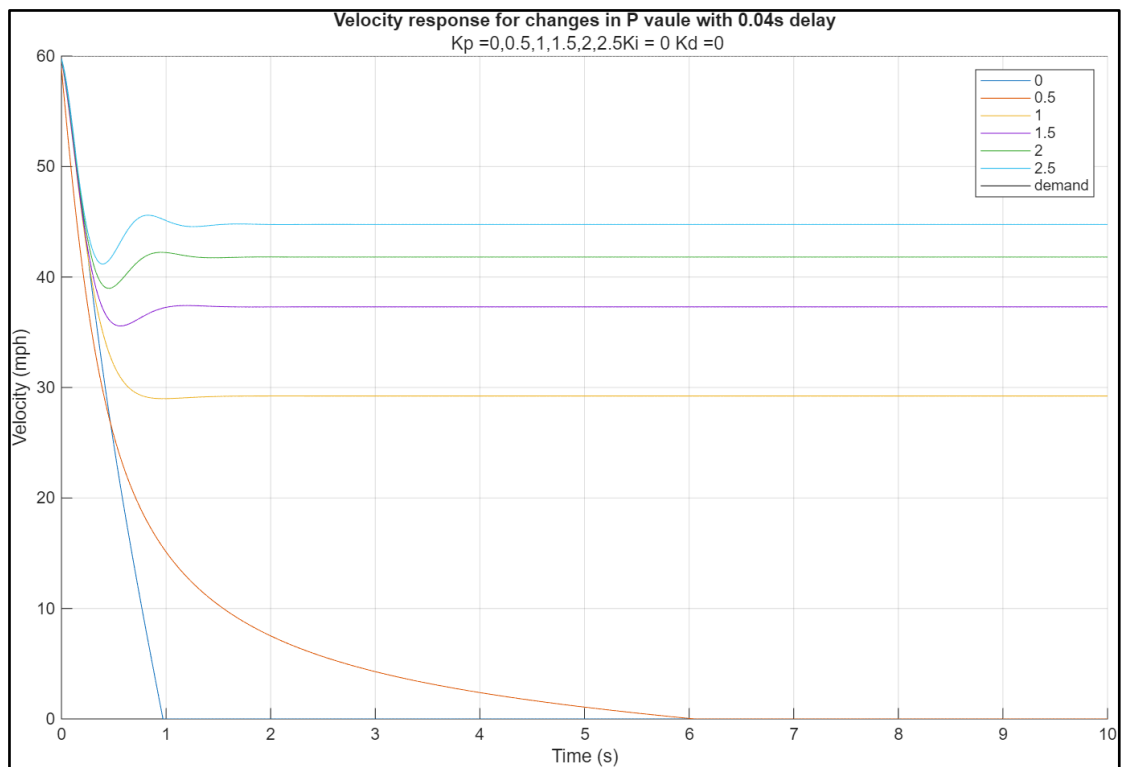


Figure 21: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 10s of Simulation

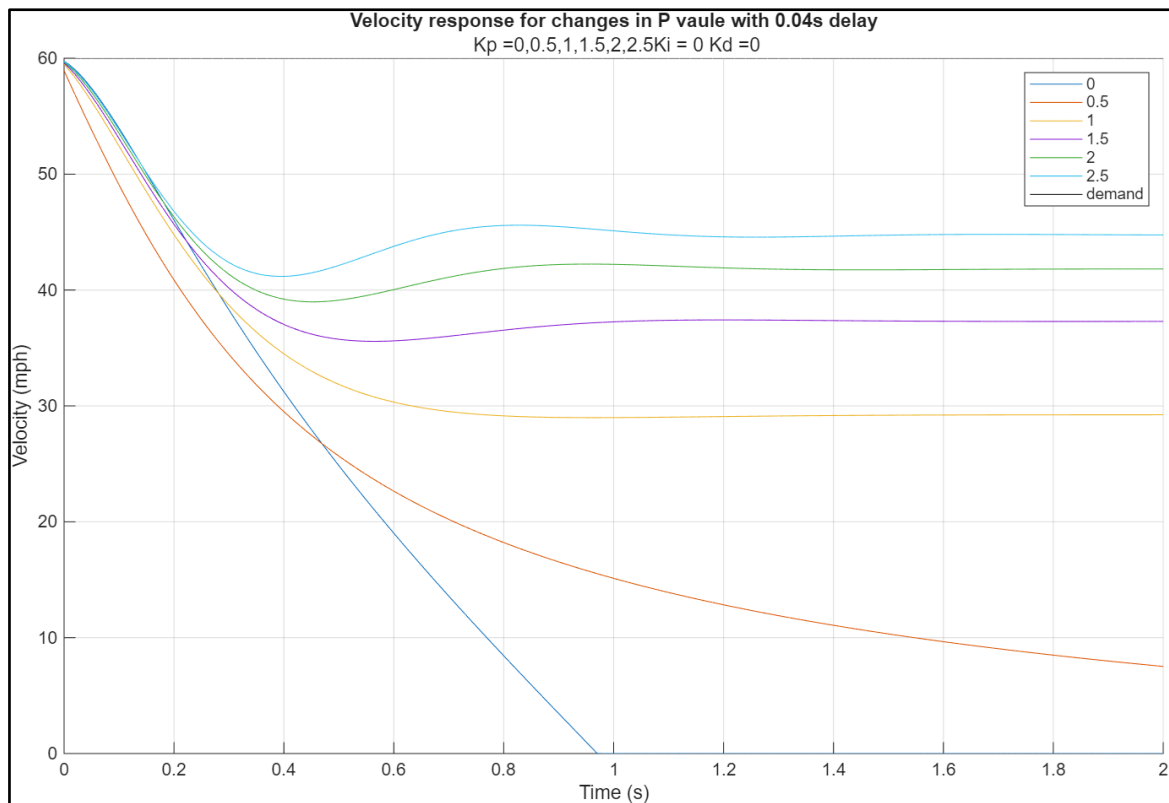


Figure 22: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 2s of Simulation

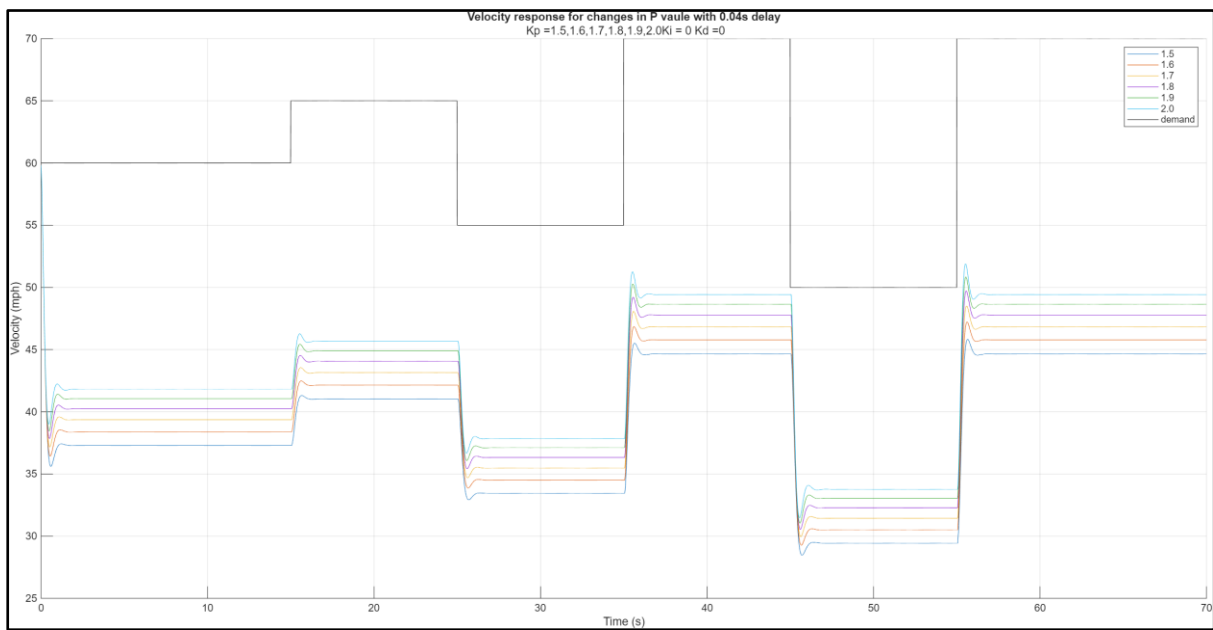


Figure 23: Graph for Further P Value Tuning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)

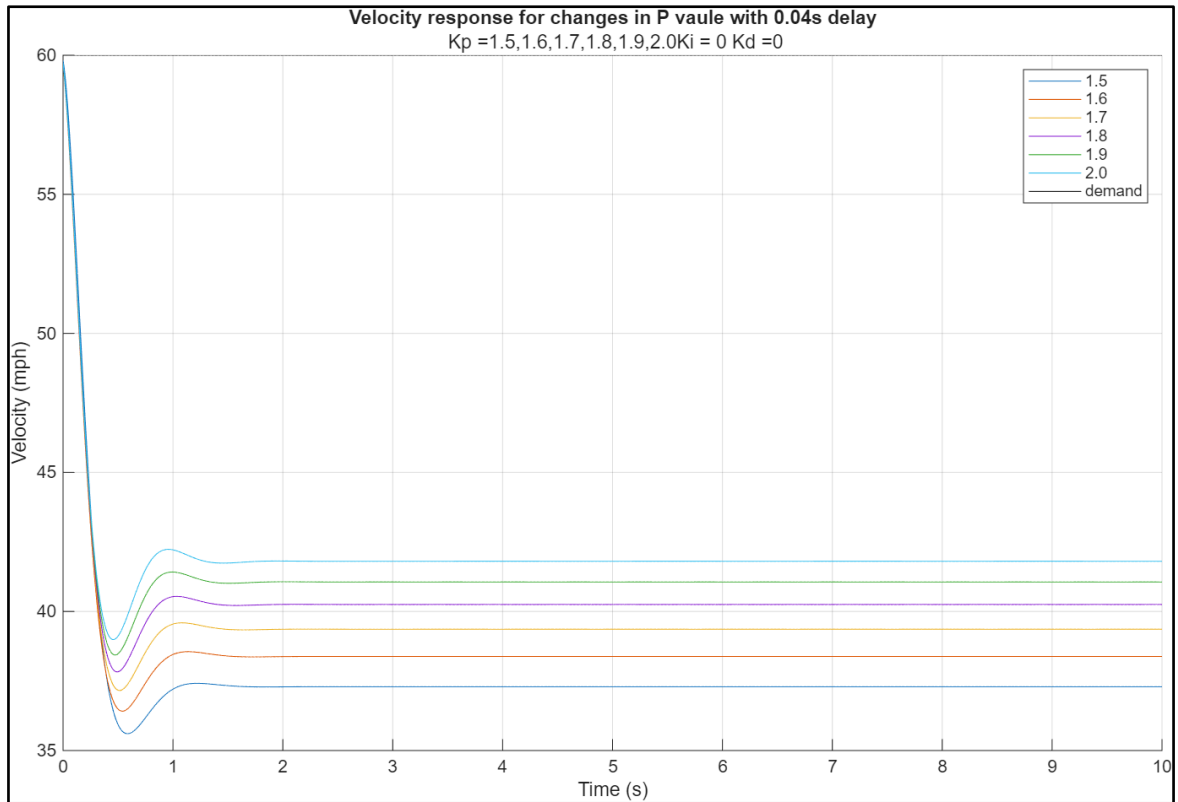


Figure 24: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 10s of Simulation

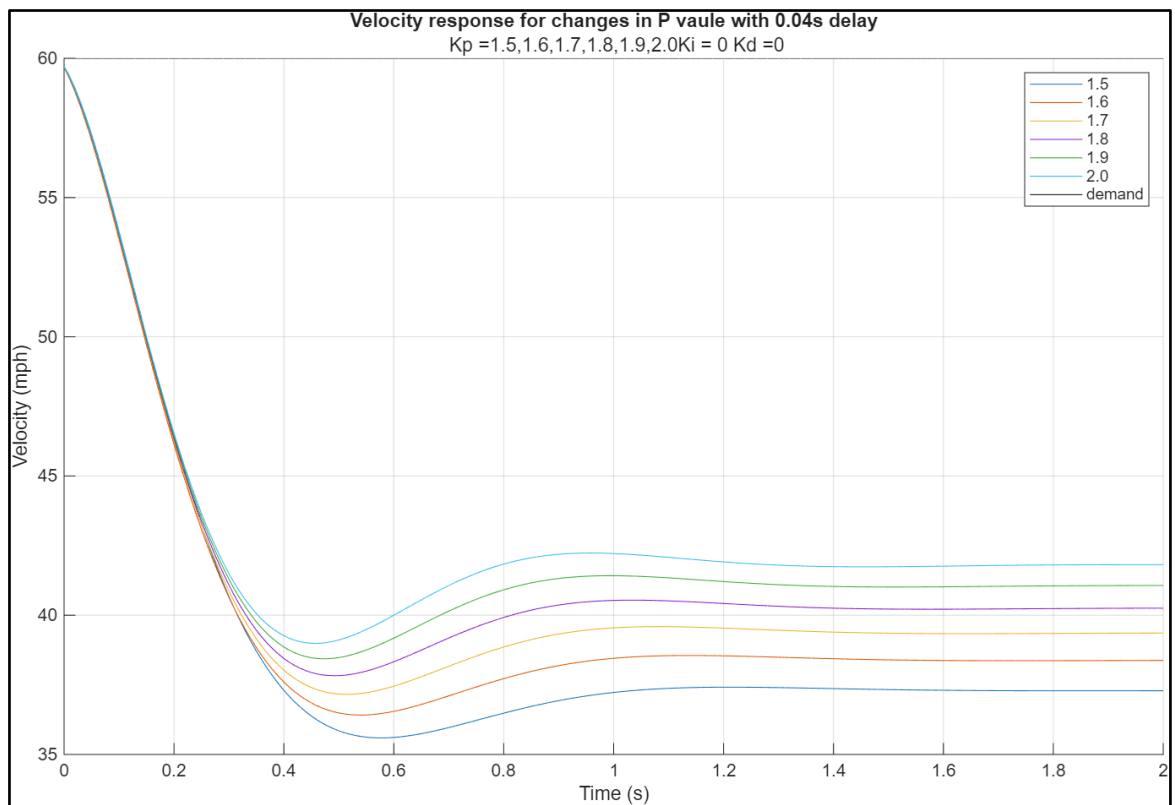


Figure 25: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 2s of Simulation (full)

4.2.2 I Tuning Results When was $P=1.7$

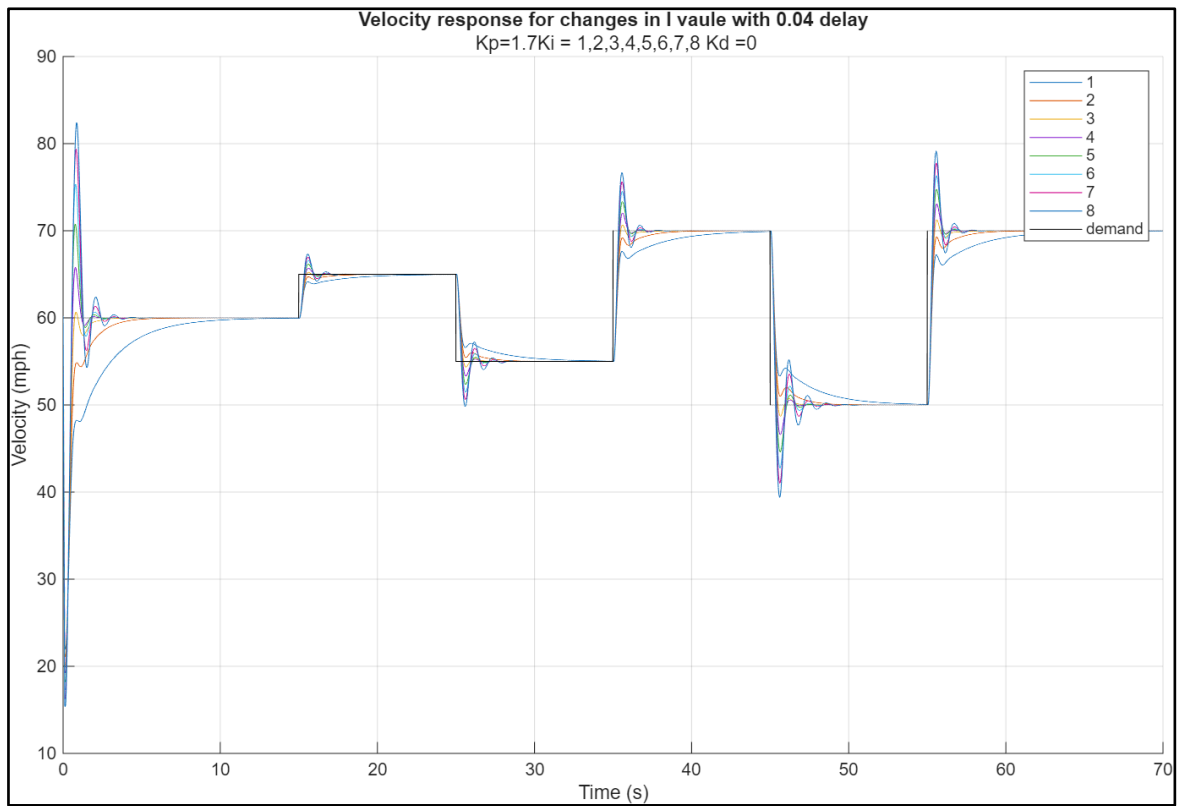


Figure 26: Graph for Initial I Value Tuning when $P=1.7$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

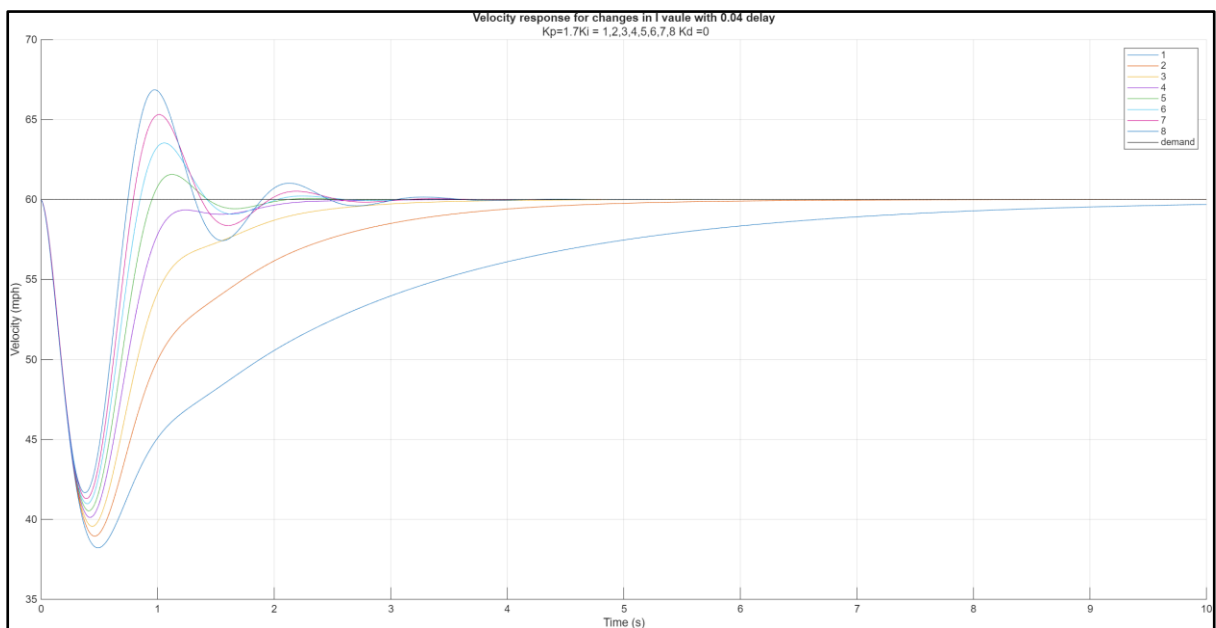


Figure 27: Graph for Initial I Value Tuning when $P=1.7$, $D=0$ and Delay is 0.04s, 10s of Simulation

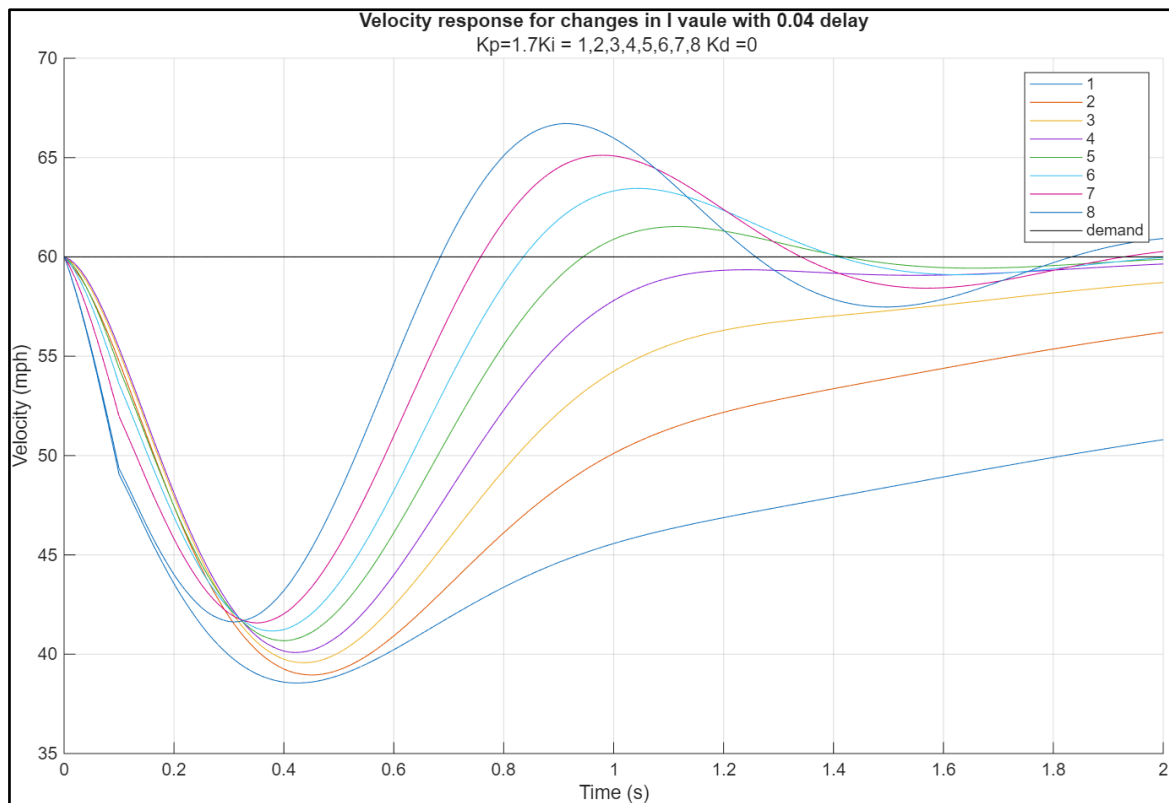


Figure 28: Graph for Initial I Value Tuning when $P=1.7$, $D=0$ and Delay is 0.04s, 2s of Simulation

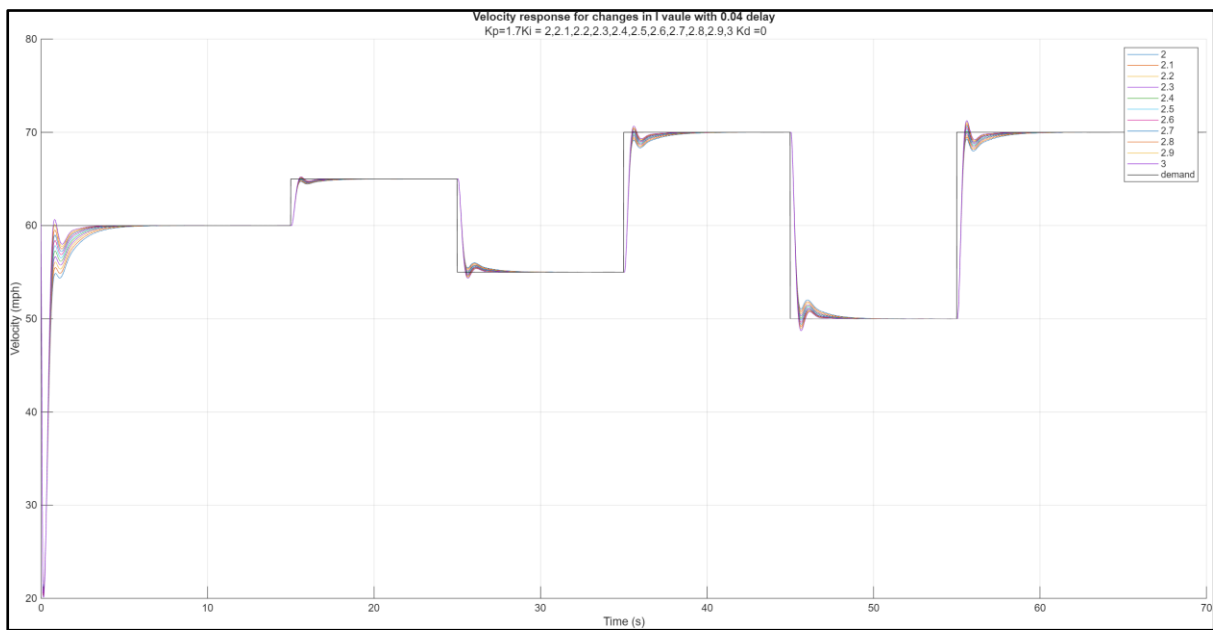


Figure 29: Graph for Further I Value Tuning when $P=1.5$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

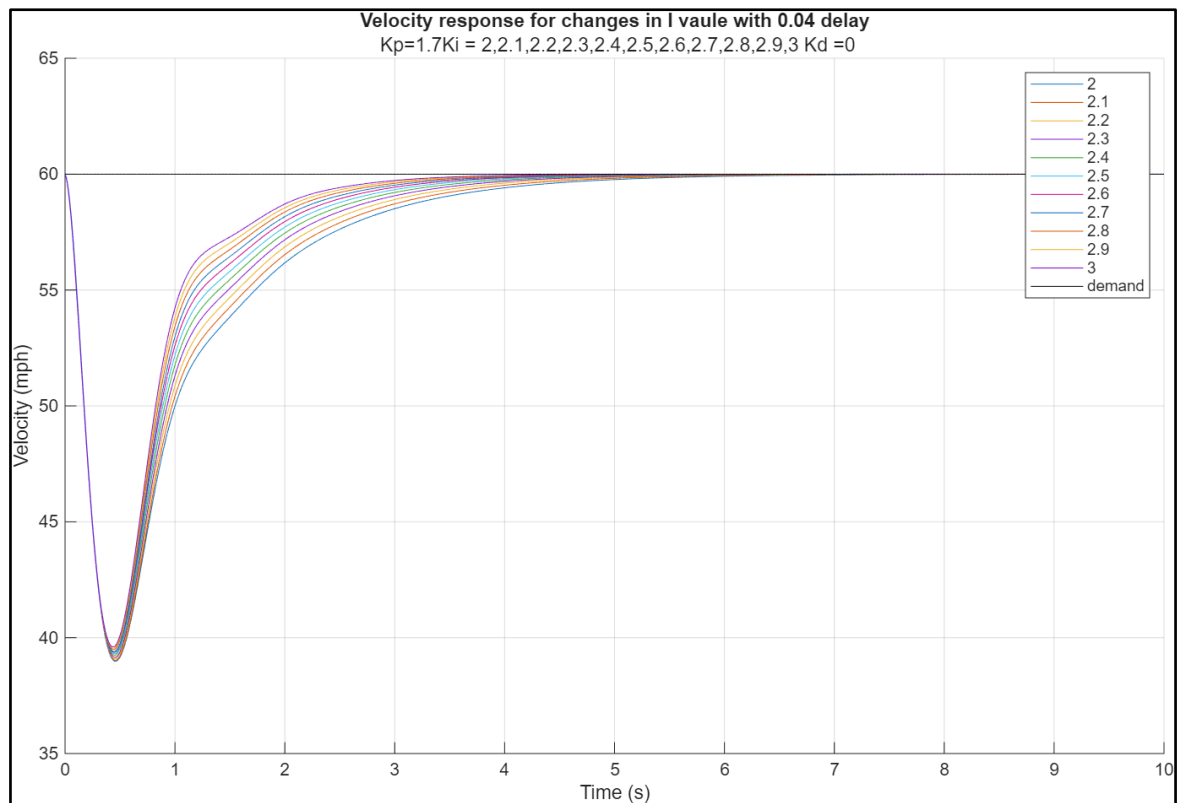


Figure 30: Graph for Further I Value Tuning when $P=1.7$, $D=0$ and Delay is 0.04s, 10s of Simulation

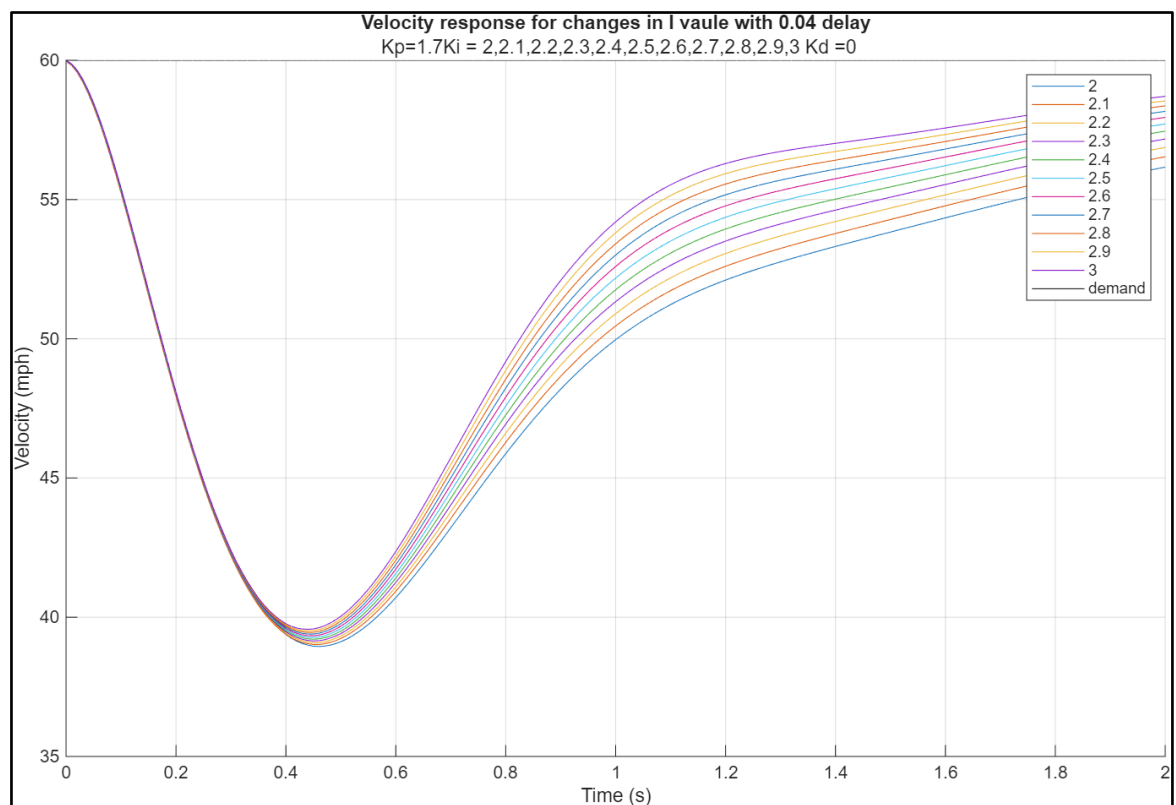


Figure 31: Graph for Further I Value Tuning when $P=1.7$, $D=0$ and Delay is 0.04s, 2s of Simulation

4.2.3 D Tunning Results When was $P=1.5$ and $I=2.6$

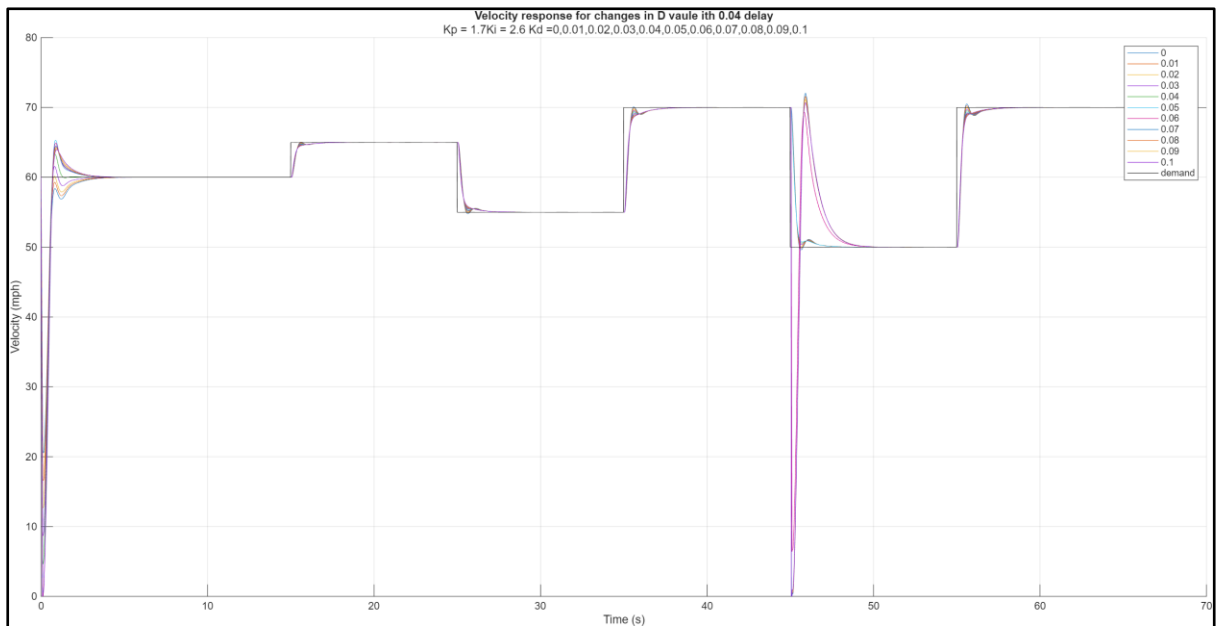


Figure 32: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 70s of Simulation (full)

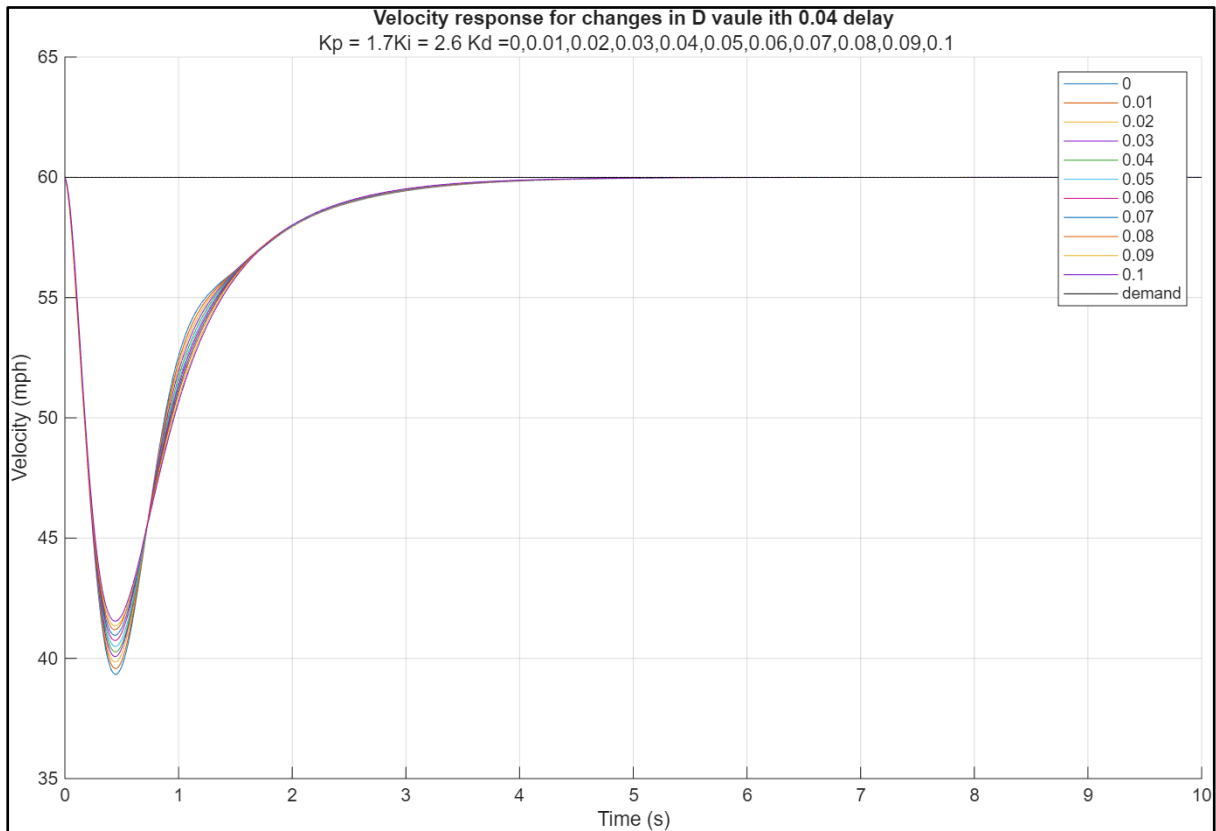


Figure 33: Graph for D Value Tunning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 10s of Simulation

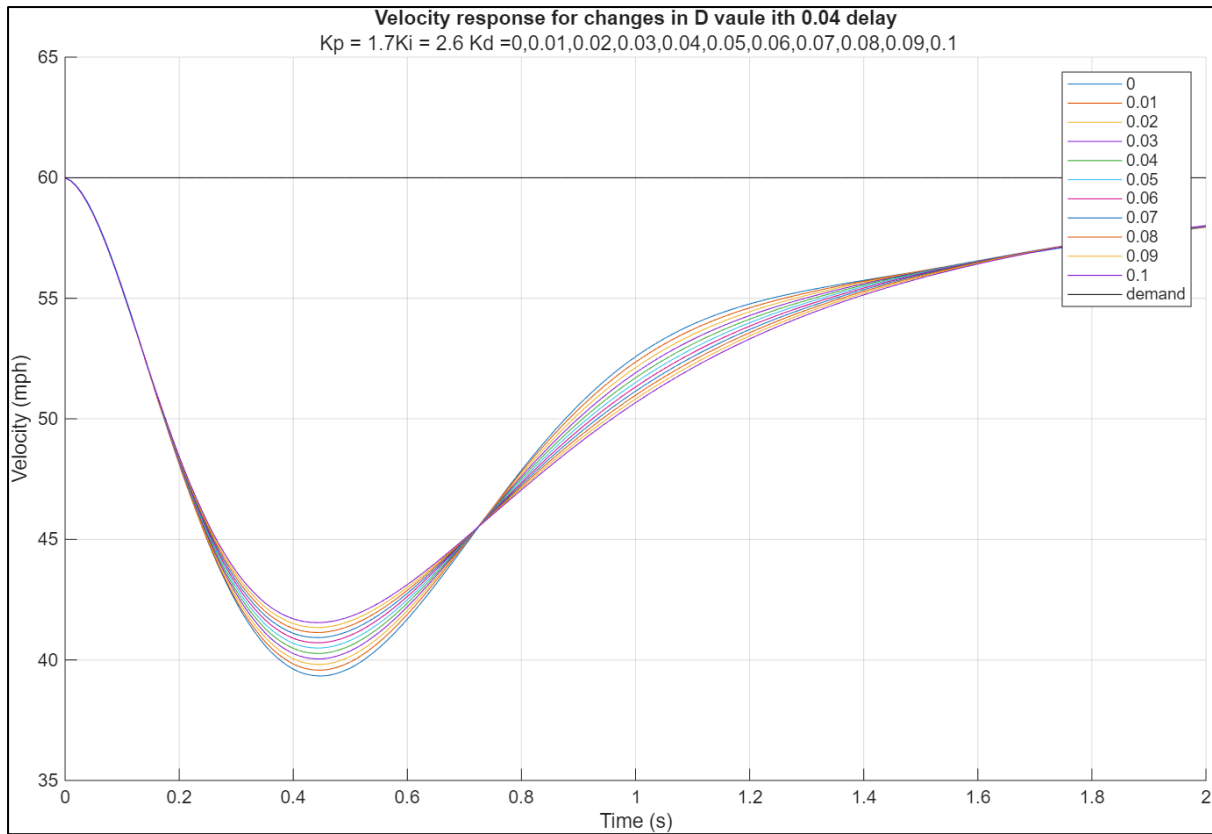


Figure 34: Graph for D Value Tuning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 2s of Simulation

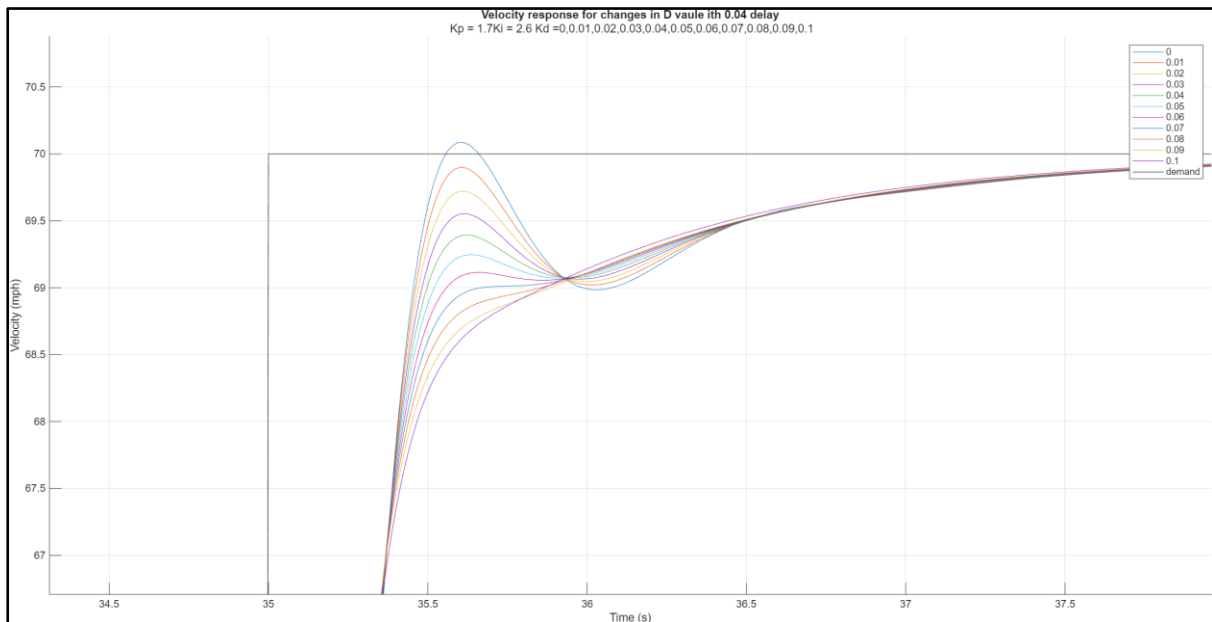


Figure 35: Graph for D Value Tuning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 34.5s to 37.5s of Simulation Close Up

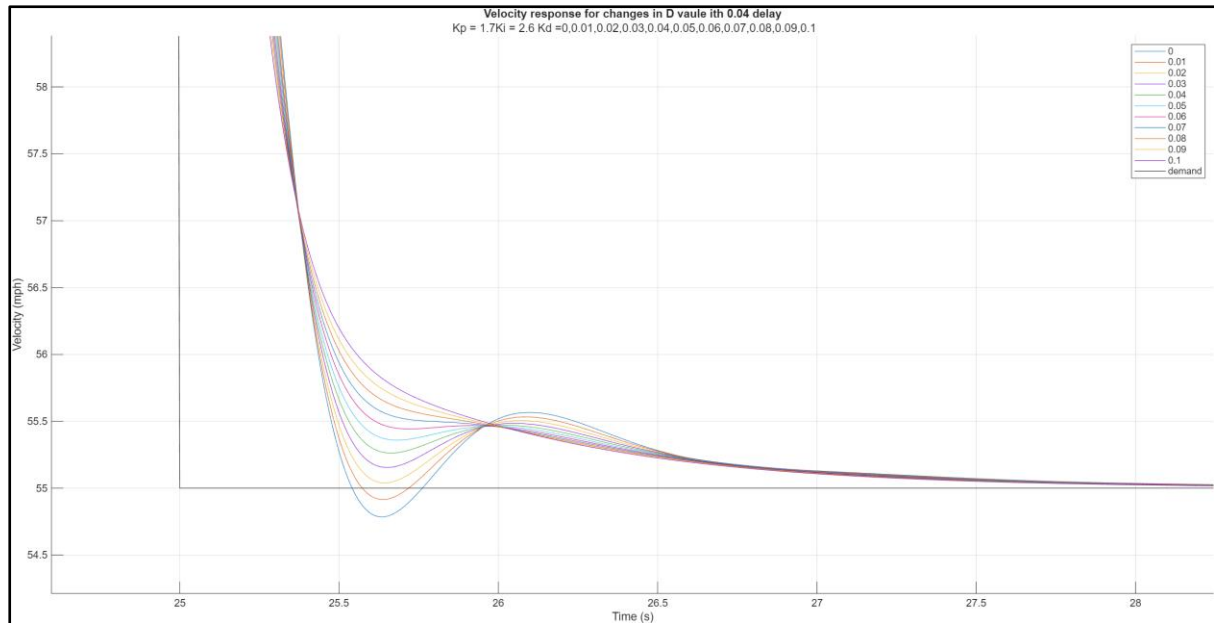


Figure 36: Graph for D Value Tuning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 25s to 28s of Simulation Close Up

4.2.4 Overview Results When $P=1.7$, $I=2.6$ and $D=0$

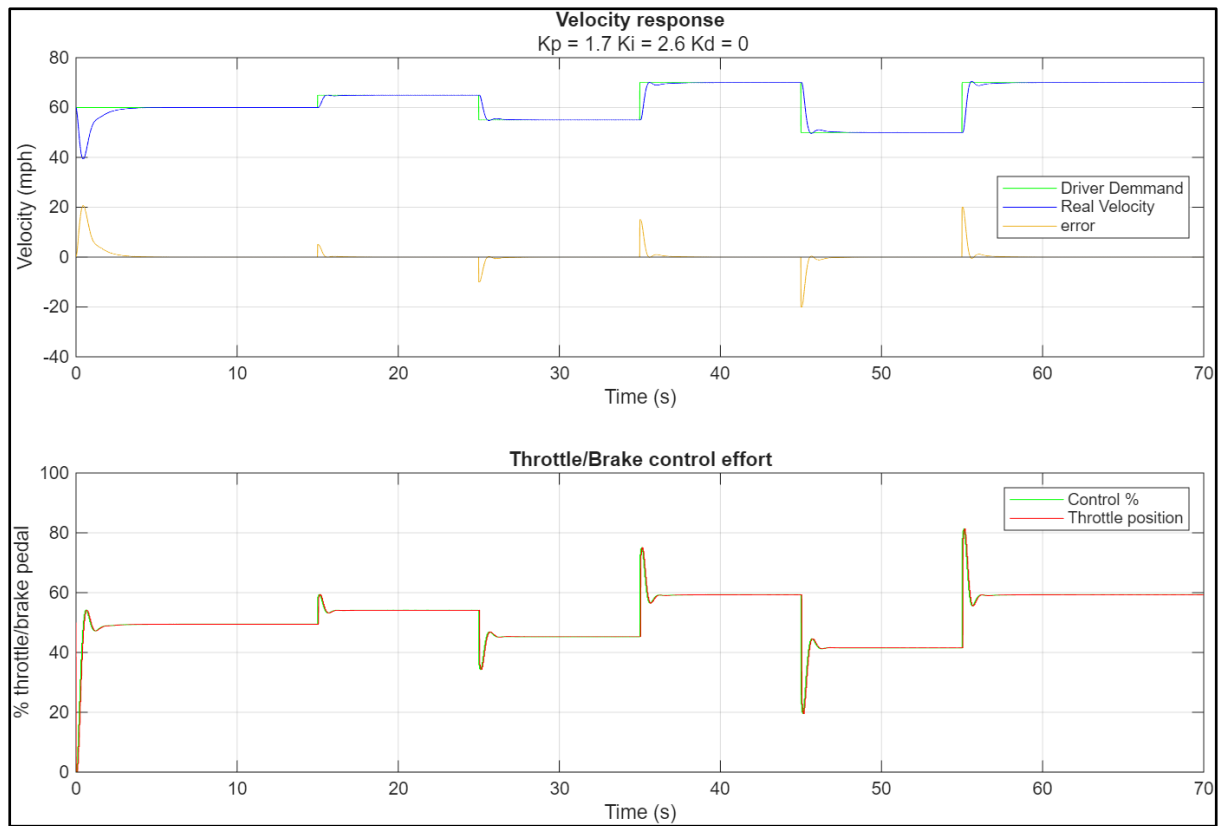


Figure 37: Graph for Overview when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

4.2.5 Bit-depth Variation Results

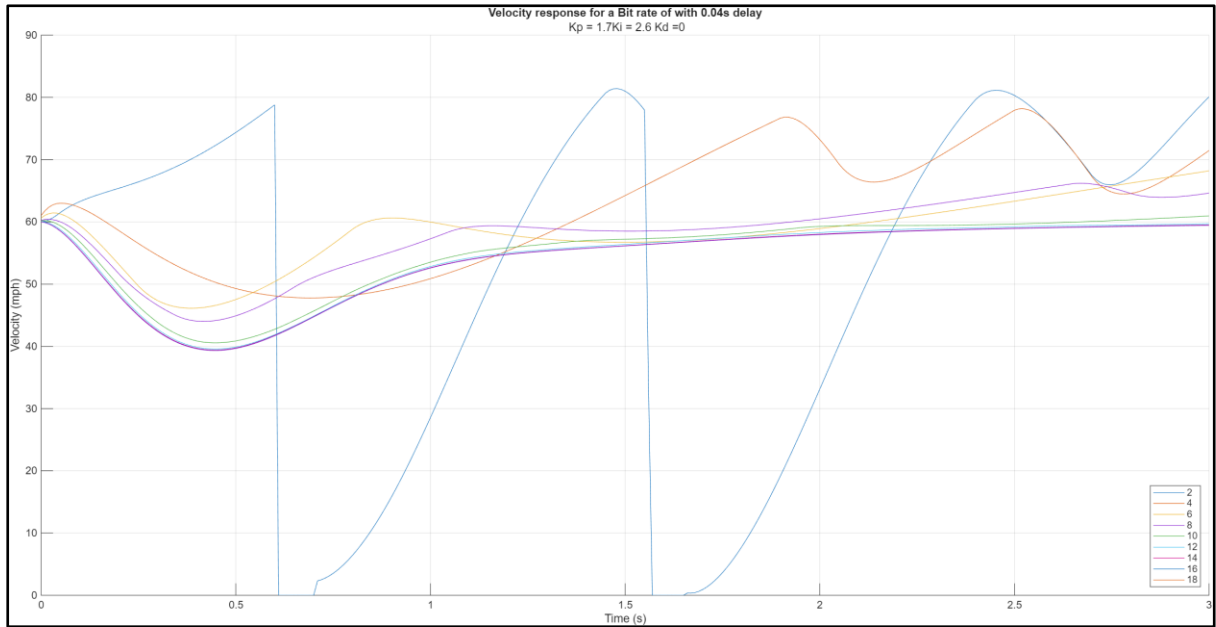


Figure 38: Graph for Bit-Depth Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 3s of Simulation

4.2.6 Sampling Time Variation Results

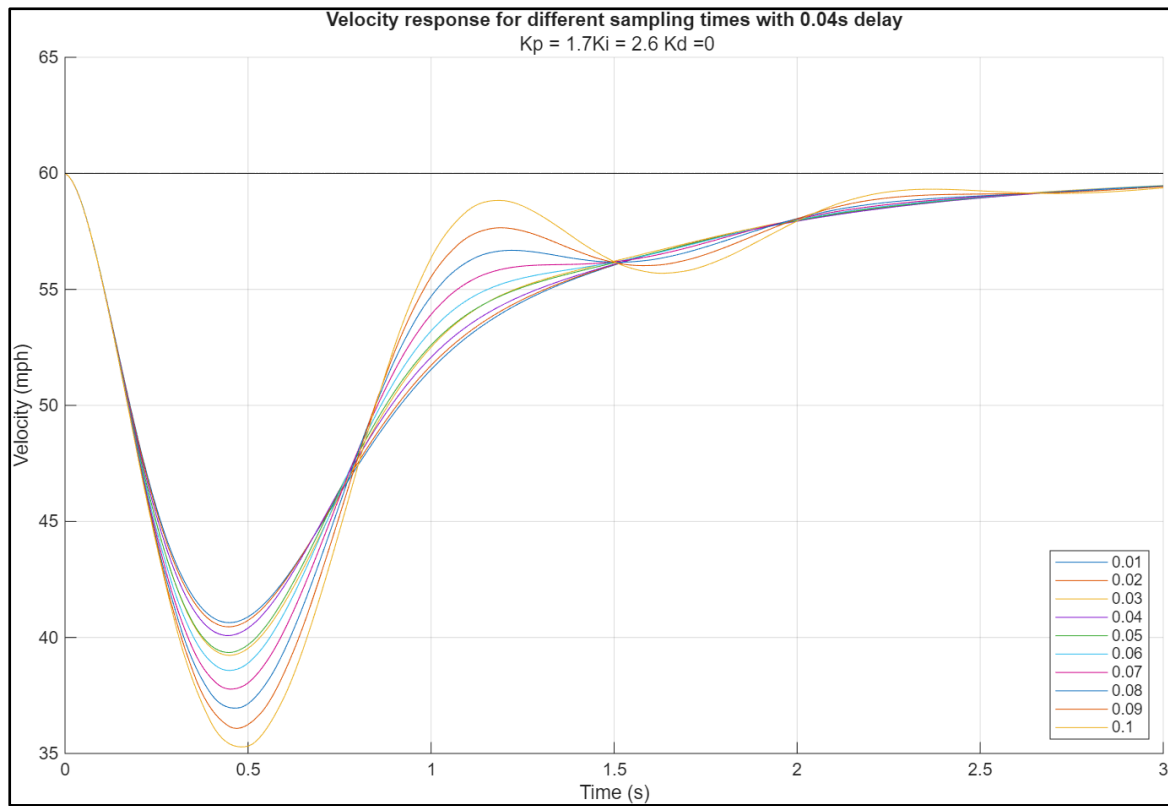


Figure 39: Graph for Sampling Time Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 3s of Simulation

5.0 Analysis

Table 4: P, I and D Values for the Two Delay Studies

Delay (seconds)	P	I	D
0.02	1.5	3.5	0.06
0.04	1.7	2.6	0

5.1 Analysis for 0.02s Delay

5.1.1 P Tuning Analysis

Increasing the proportional gain decreased the response time and decreases the steady-state error; however, higher values (e.g. $P=4$) introduced significant overshoot and oscillations (Figure 1, 2 and 3). This behaviour indicates that the system is less stable compared to when P is lower. 1.5 was selected for the P value as it provided a fast response time without overly destabilising the demand signal. (Figure 4, 5 and 6).

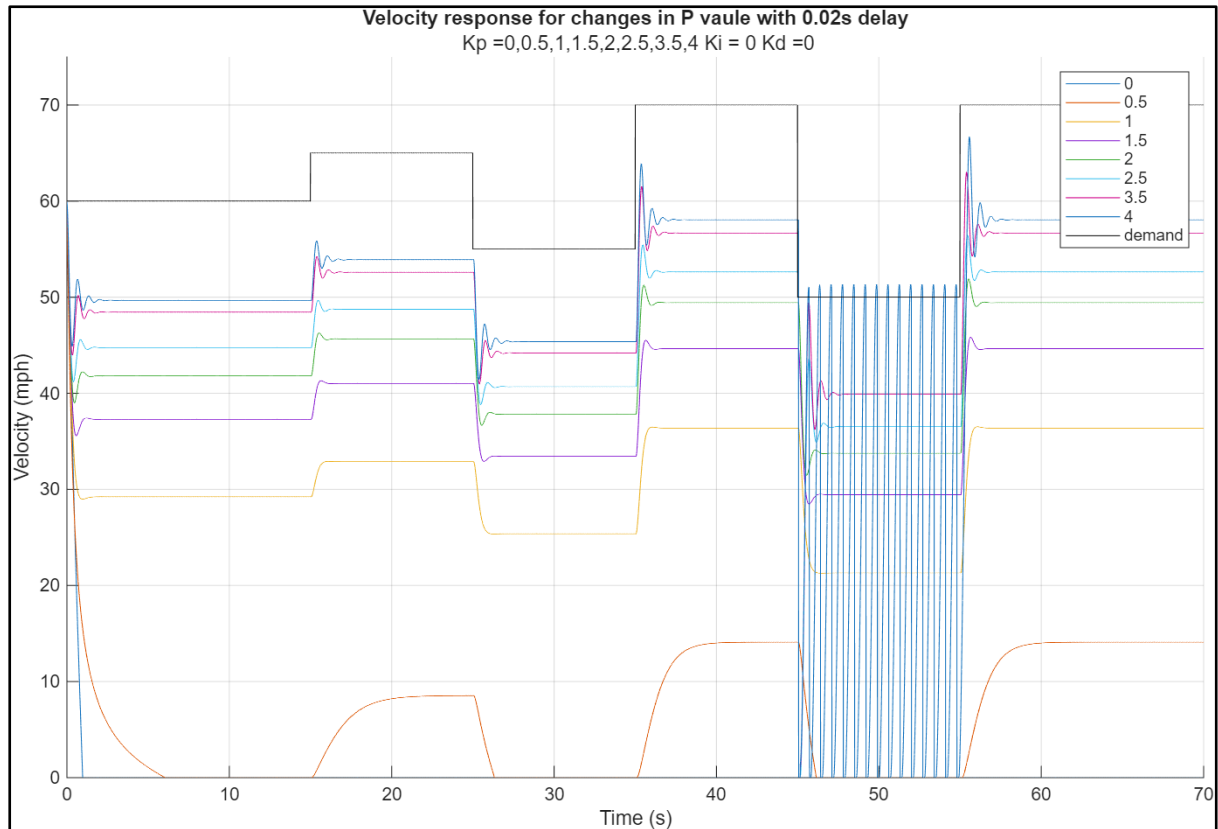


Figure 1: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.02s, 70s of Simulation (full)

5.1.2 I Tuning Analysis

Integral gain reduced the steady-state error by accumulating the error over time (Figure 7, 8, 9, 10, 11, 12 and 13). However, increasing the I value also resulted in increased overshoot and longer settling times due to integral windup. A value of $I = 3.5$ was selected as it effectively minimised steady-state error without introducing excessive instability.

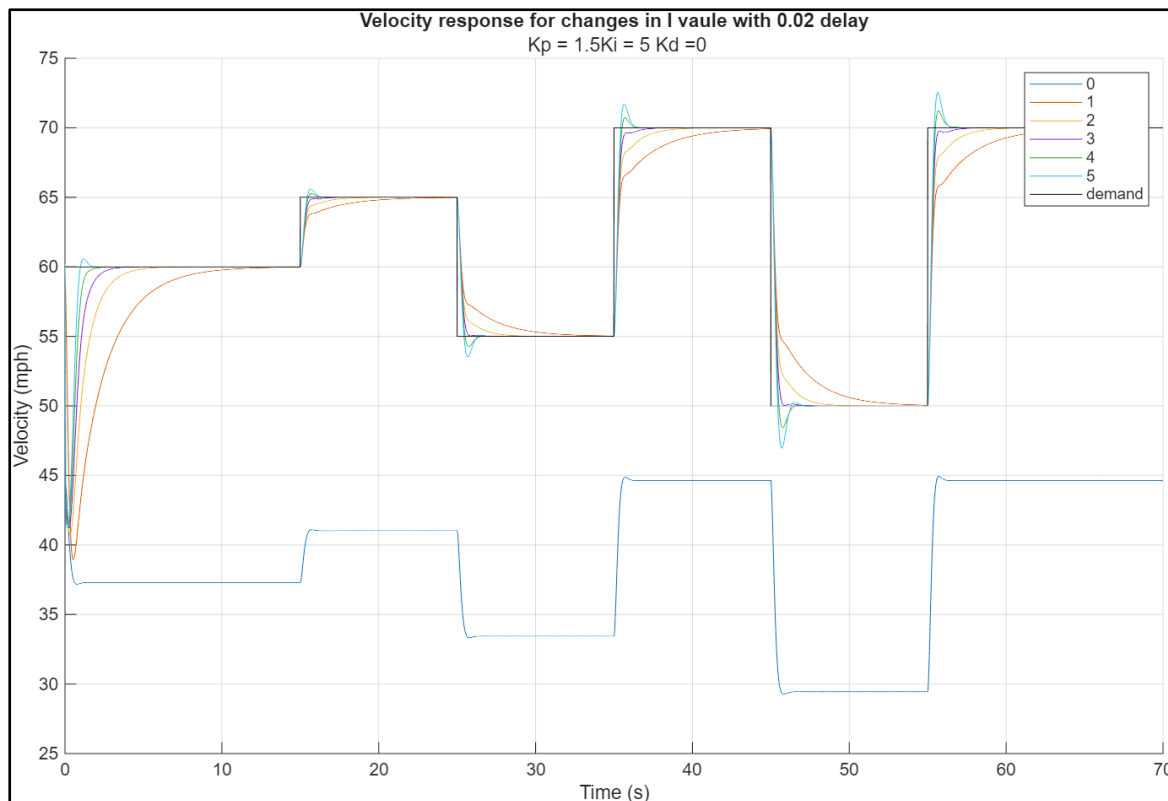


Figure 7: Graph for Initial I Value Tuning when $P=1.5$, $D=0$ and Delay is 0.02s, 70s of Simulation (full)

5.1.3 D Tuning Analysis

The derivative gain was used to reduce overshoot, settling time and dampen oscillations (Figure 14, 15 and 16). Increasing D improved system stability and reduced peak overshoot; however, excessive derivative gain can introduce noise sensitivity. When $D = 0.06$ the damping was sufficient without negatively impacting system response.

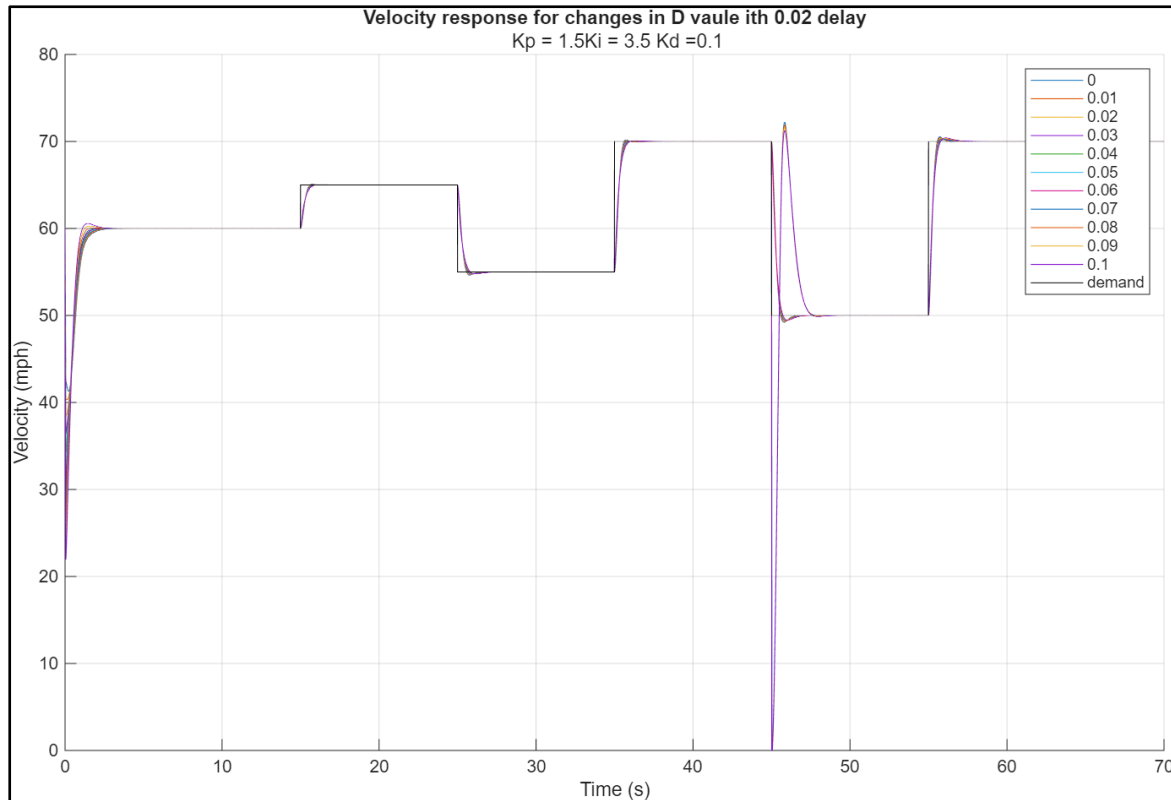


Figure 14: Graph for D Value Tuning when $P=1.5$, $I=3.5$ and Delay is 0.02s, 70s of Simulation (full)

5.1.4 Overview Analysis

Overall, the tuned PID controller demonstrated stable and responsive behaviour at a delay of 0.02s, with low settling time and minimal overshoot. The system was able to track the demand signal effectively, indicating that the controller was adequately tuned. However, an anomaly was observed at the start of the simulation, where the vehicle velocity briefly drops to 0 mph despite an initial condition of 60 mph. This behaviour is likely due to errors within the simulation model, as such rapid deceleration would be unrealistic in a physical system. This highlights a limitation in the model implementation rather than the controller design.

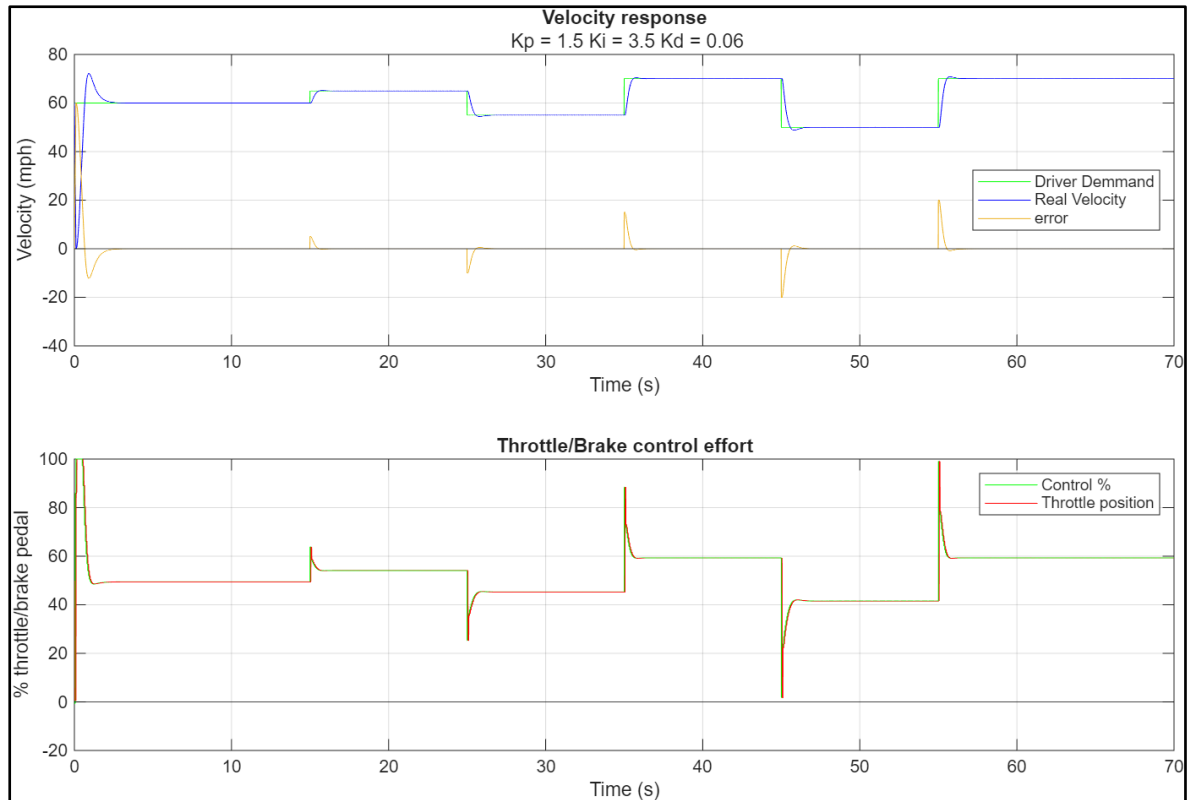


Figure 17: Graph for Overview when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 70s of Simulation (full)

5.1.5 Bit-depth Variation Analysis

Increasing the bit-depth improves the resolution of the control signal (Figure 18). A bit-depth of 12 or more results in a smooth system response. Values less than 12 reduced accuracy and caused fluctuations in velocity. This demonstrates the importance of sufficient signal resolution to maintain accuracy and stability.

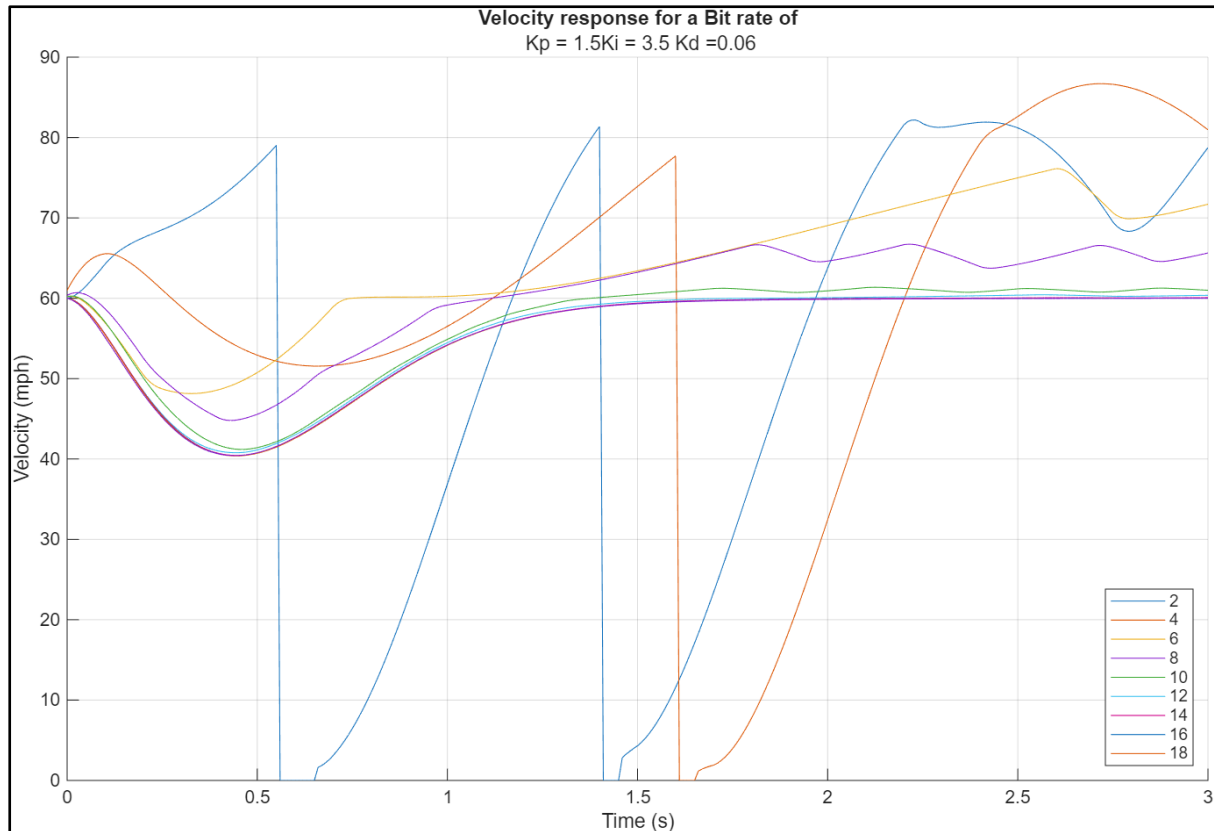


Figure 18: Graph for Bit-Depth Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation

5.1.6 Sampling Time Variation Analysis

The lower the sampling time the more frequent the communication is to the PID controller, this reduces the overshoot and settling time (Figure19). However, excessively small times requires more computational demand and may induce noise. Larger times reduce the accuracy and response time of the system thus a compromise must be made between accuracy and the computational demand.

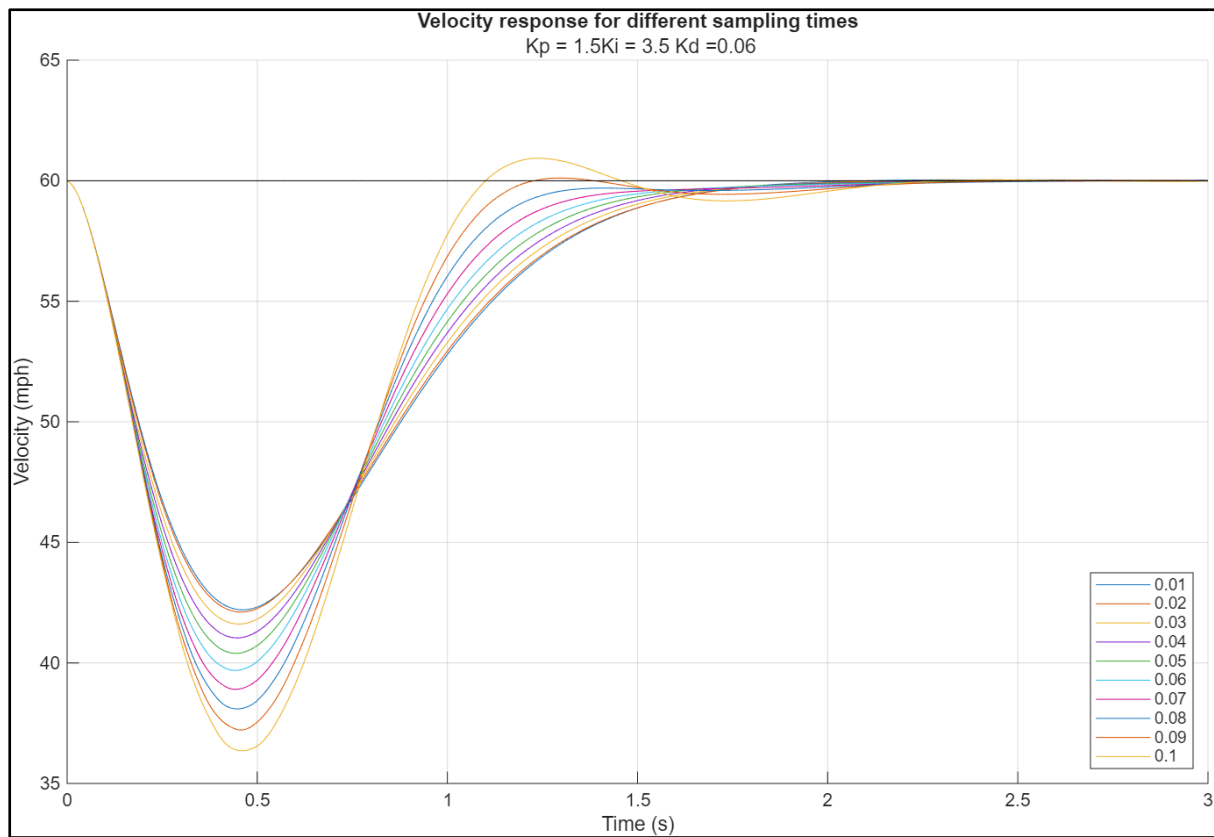


Figure 19: Graph for Sampling Time Variation when $P=1.5$, $I=3.5$, $D=0.06$ and Delay is 0.02s, 3s of Simulation

5.2 Analysis for 0.04s Delay

5.2.1 P Tuning Analysis

Figure 20, 21 and 22 show the initial P values and Figure 23, 24 and 25 show the further tuning. Increasing the delay increased the error and caused the signal to be more prone to overshoot. $P=1.7$ was selected.

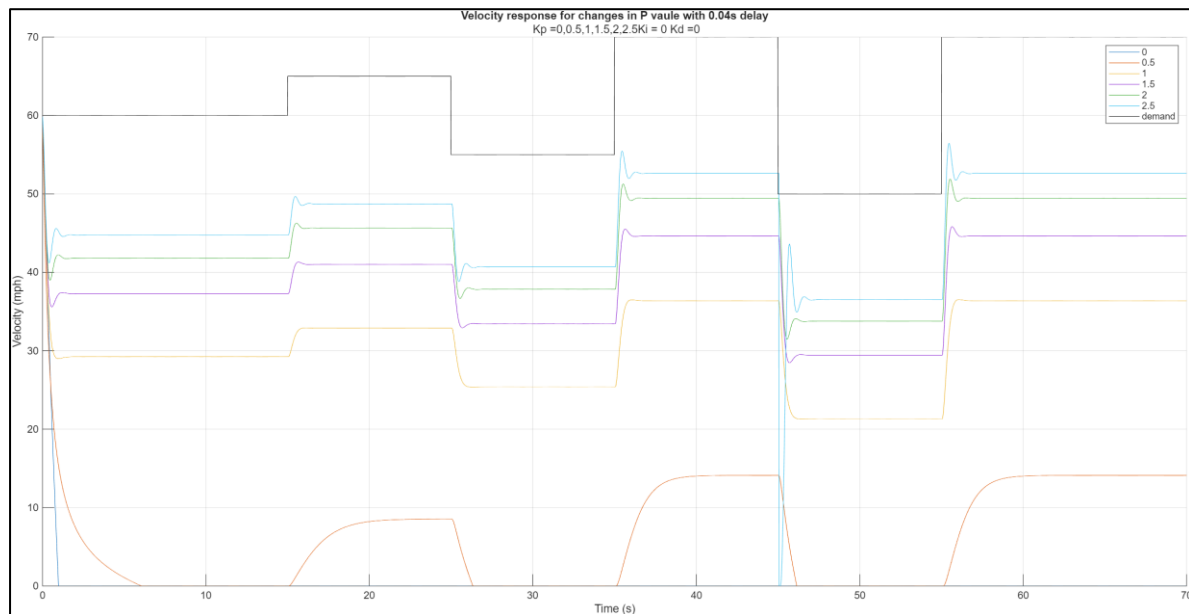


Figure 20: Graph for Initial P Value Tuning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)

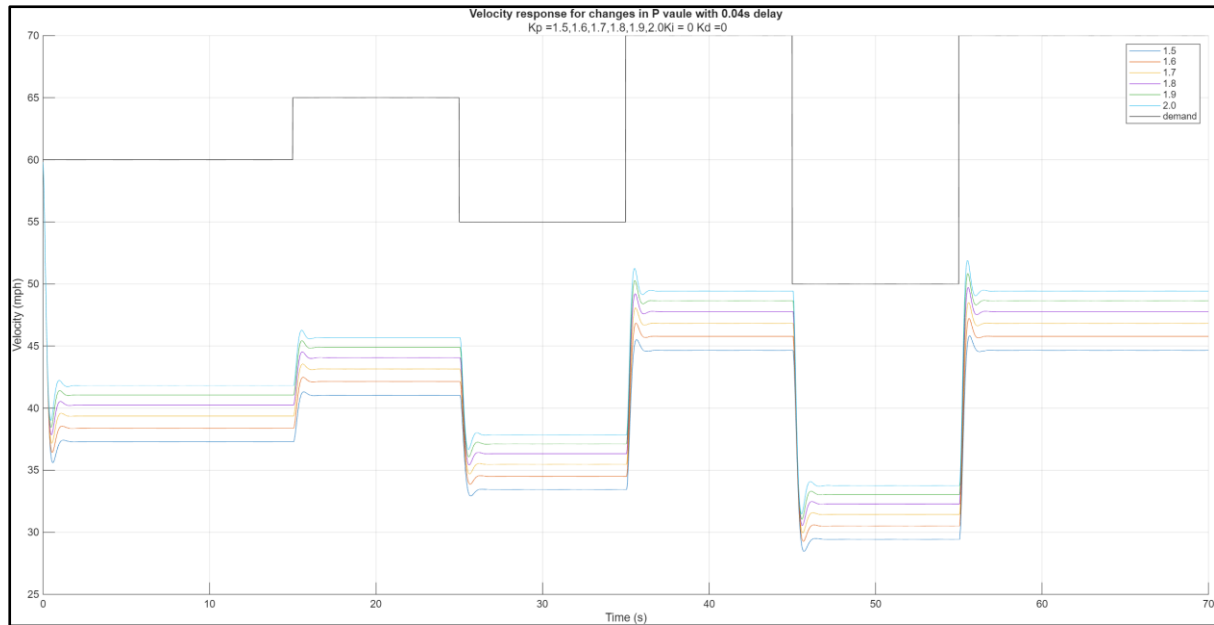


Figure 23: Graph for Further P Value Tuning when I and D are 0 and Delay is 0.04s, 70s of Simulation (full)

5.2.2 I Tuning Analysis

Increasing the I value resulted in increased overshoot, longer settling times and significant oscillation due to integral windup (Figure 26, 27 and 28). $I = 2.6$ was selected as it effectively minimised steady-state error without introducing excessive instability (Figure 29, 30 and 31).

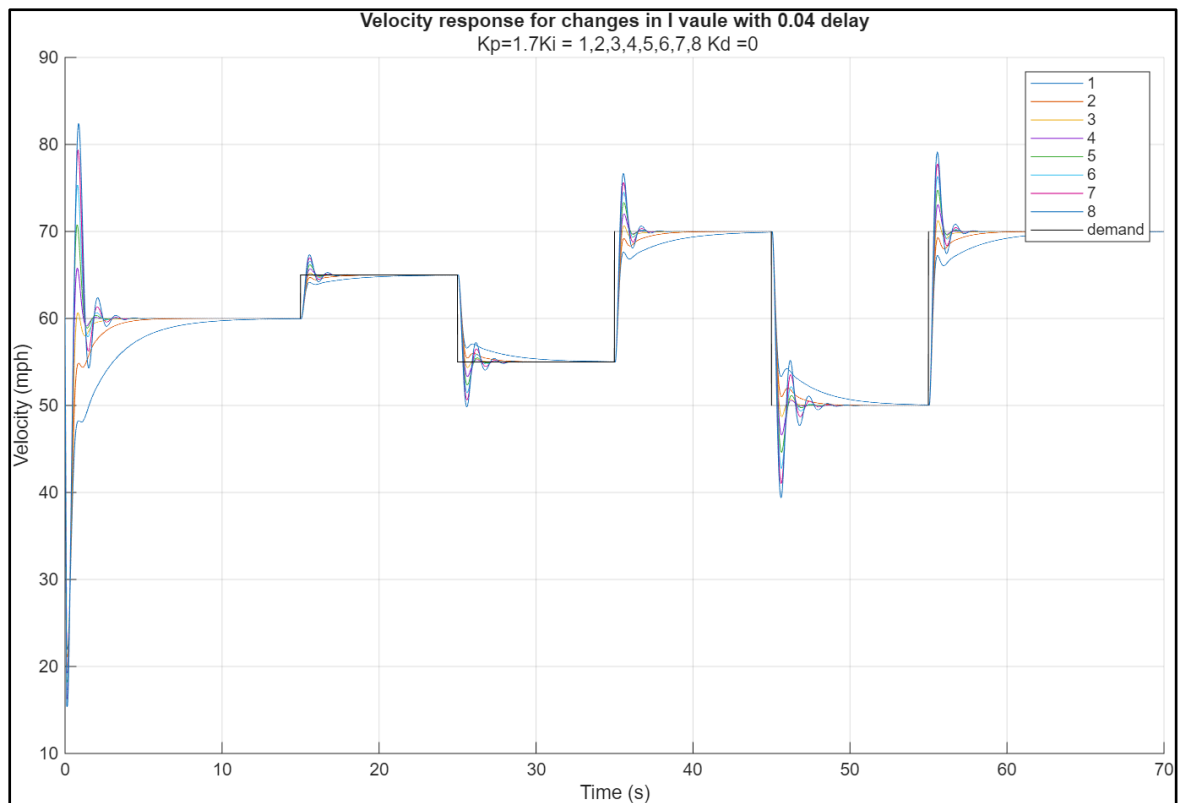


Figure 26: Graph for Initial I Value Tunning when $P=1.7$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

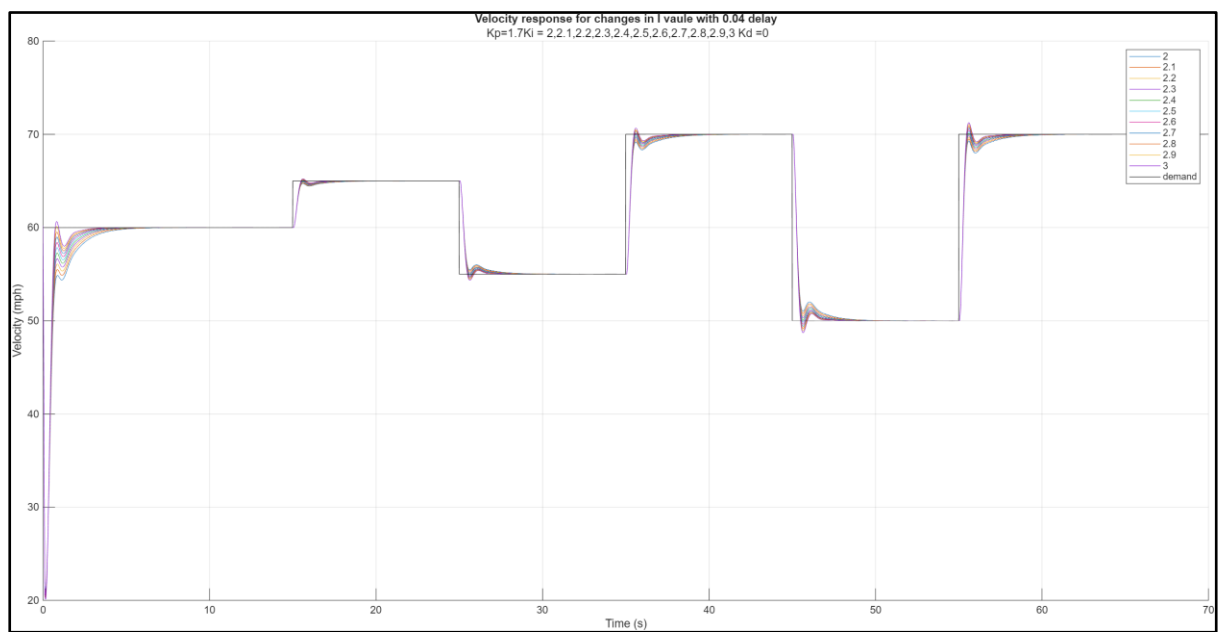


Figure 29: Graph for Further I Value Tunning when $P=1.5$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

5.2.3 D Tuning Analysis

Increasing D reduced peak overshoot; however, excessive derivative gain can introduce noise sensitivity (Figure 32, 33 and 34). When $D = 0$ the system had a small overshoot which can be desirable. On reflection a D value of 0.05 may be more appropriate as there is no overshoot and the settling time is very similar to when $D=0$.

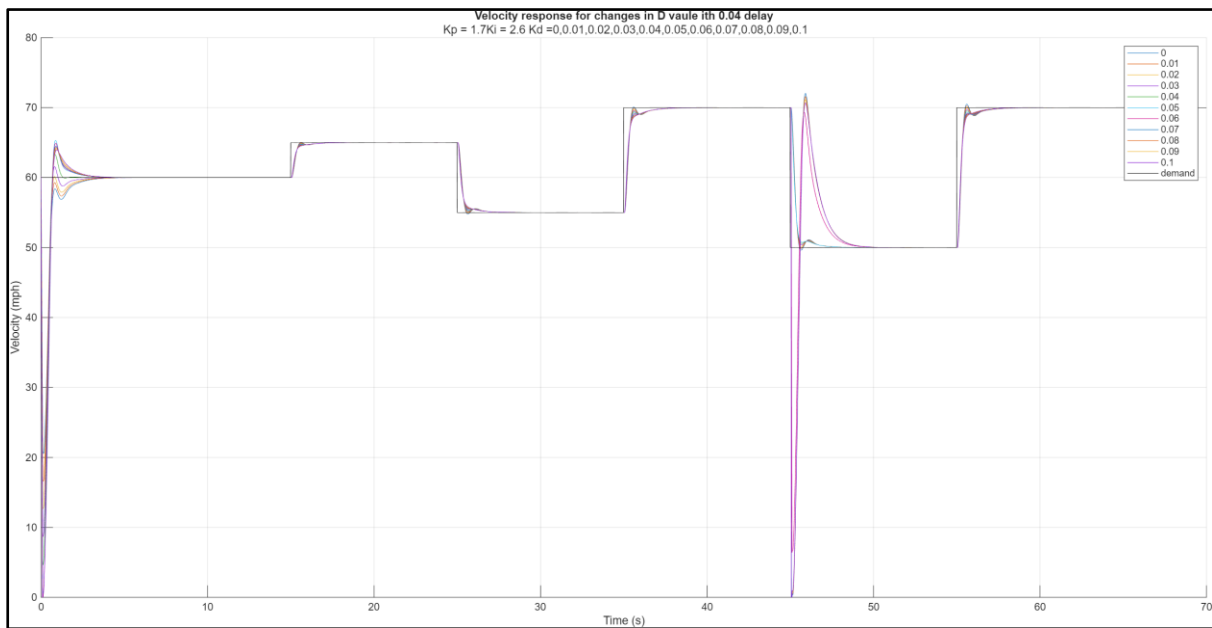


Figure 32: Graph for D Value Tuning when $P=1.7$, $I=2.6$ and Delay is 0.04s, 70s of Simulation (full)

5.2.4 Overview Analysis

The increase in delay to 0.04s significantly reduced the controller performance. The controller exhibited slower response time, increased overshoot and reduced stability, highlighting the reduction in performance due to the delay. The greater delay introduced a lag between the velocity and response, making the system more difficult to tune. Additionally, the controller relied more heavily on braking rather than throttle input, indicating inefficiencies in control behaviour. The initial braking observed at the start of the simulation further supports the theory of the model limitations.

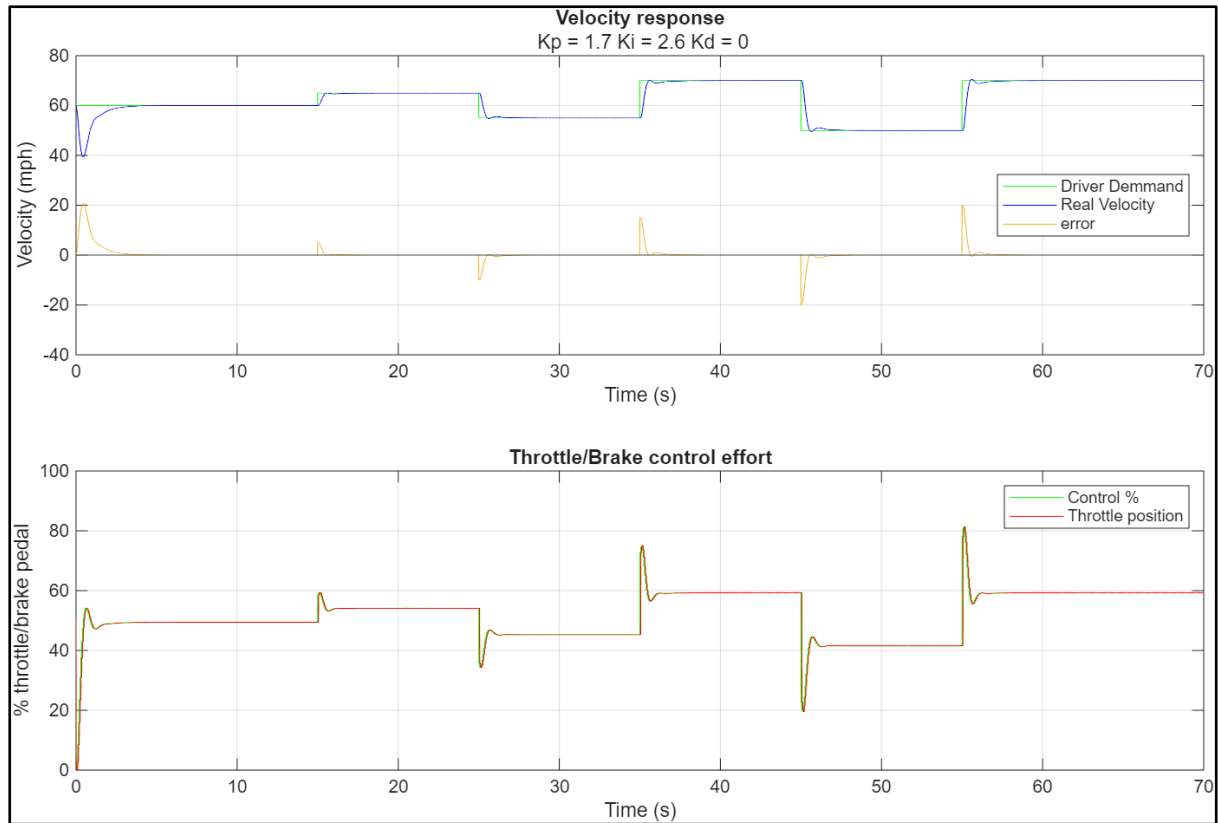


Figure 37: Graph for Overview when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 70s of Simulation (full)

5.2.5 Bit-depth Variation Analysis

The effects of bit-depth changes were more pronounced at 0.04s delay . When the delay was 0.02s the lowest acceptable bit-depth was 12. While that 0.04s delay a bit-depth of 14 or greater would be advisable to counter act the instability caused by the increased delay.

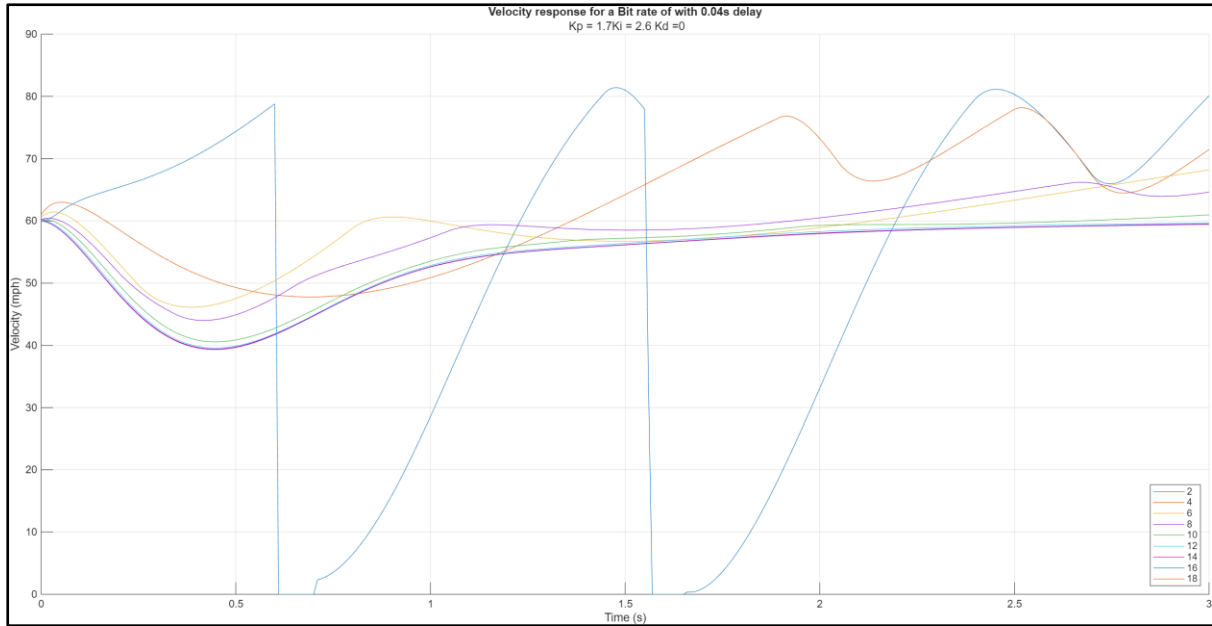


Figure 38: Graph for Bit-Depth Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is $0.04s$, $3s$ of Simulation

5.2.6 Sampling Time Variation Analysis

The effects of sampling time was more pronounced at the higher delay value. This resulted in greater oscillations and longer settling times. When the delay was $0.02s$ all velocity values had settled by 2 seconds opposed $2.5s$ when the delay was $0.04s$. This demonstrates that the increase in delay amplified the negative effects of larger times, further reducing control accuracy and system stability.

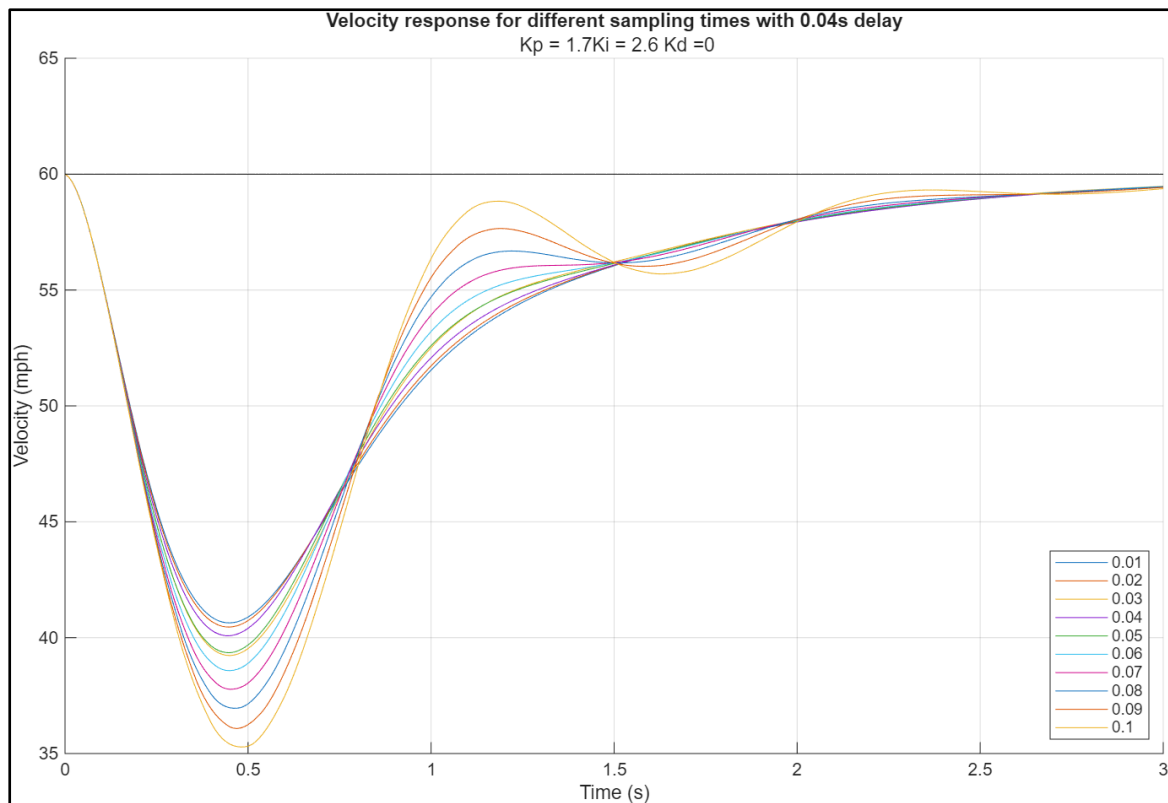


Figure 39: Graph for Sampling Time Variation when $P=1.7$, $I=2.6$, $D=0$ and Delay is 0.04s, 3s of Simulation

6.0 Discussion & Conclusions

6.1 Model Weaknesses

The vehicle model is a source of error within the project as it is not an accurate simulation of true vehicle behaviour.

6.2 Conclusions

The analysis demonstrated that delay has a significant impact on controller performance. Lower delay values resulted in improved stability, reduced response time, and more effective tuning. While increased delay values reduced controller performance by introducing lag, increasing overshoot and oscillations. This highlights the importance of minimising delay in real-world control systems to maintain performance.

7.0 Future work

7.1 Automated Tuning

To further improve the cruise control system an automated tuning system should be implemented. Machine learning could be used to tune the controller.

7.2 Acceleration Limits

To meet ISO standards (chapter 2.4) the vehicle must not accelerate over 3.5ms^{-2} or -2.5ms^{-2} . In Figure 40 the yellow surface represents the 3.5ms^{-2} limit and the purple represents the -2.5ms^{-2} limit. The curved surface is the vehicle acceleration for different control and velocity values. The vehicle does not meet the standards and would require additional coding of a limiter to meet the requirements as the acceleration exceeds -2.5ms^{-2} .

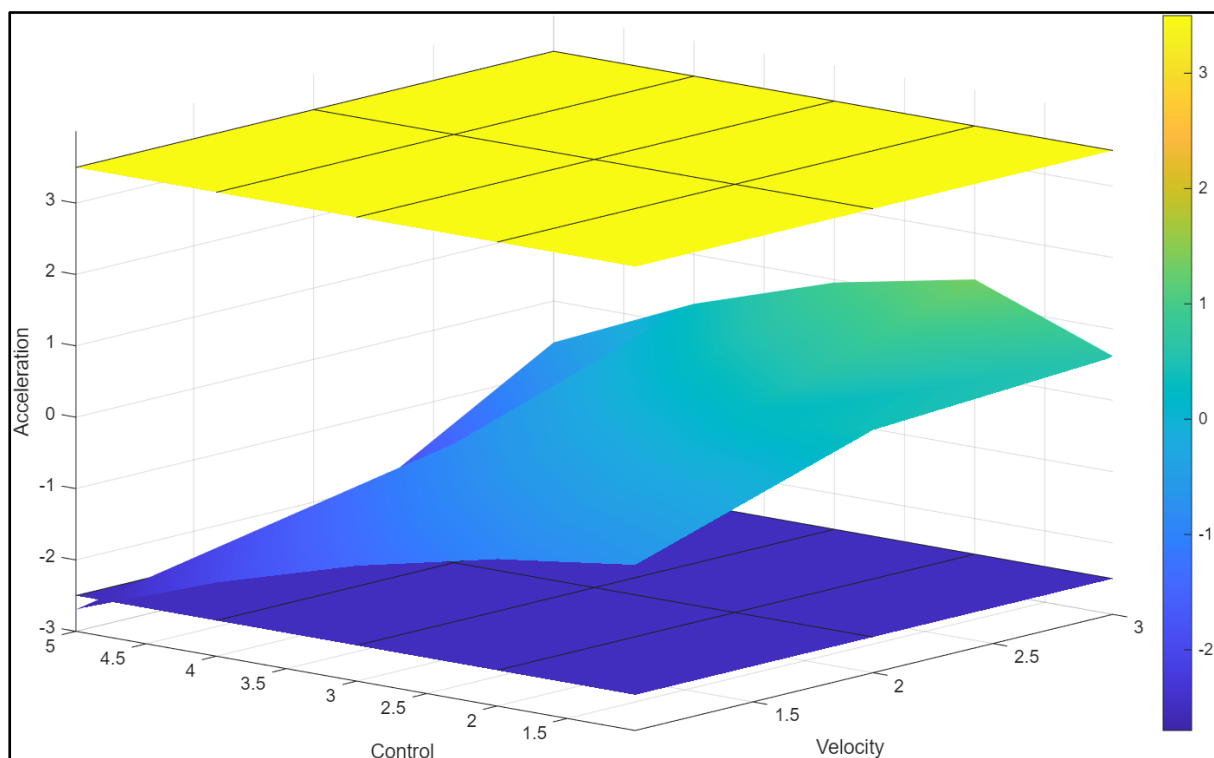


Figure 40: Graph of Acceleration Limits from ISO

8.0 References

Bennett, J., Bhasin, A., Grant, J. and Chung Lim, W. (2023) '9: Proportional-integral-derivative (PID) control', *Proportional-Integral-Derivative (PID) Control*, (9.3: PID Tuning via Classical Methods) Available at:

[https://eng.libretexts.org/Bookshelves/Industrial and Systems Engineering/Chemical Process Dynamics and Controls %28Woolf%29/09%3A Proportional-Integral-Derivative %28PID%29 Control/9.03%3A PID Tuning via Classical Methods?utm_source=](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls%28Woolf%29/09%3A_Proportional-Integral-Derivative%28PID%29_Control/9.03%3A_PID_Tuning_via_Classical_Methods?utm_source=)

[e.](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls%28Woolf%29/09%3A_Proportional-Integral-Derivative%28PID%29_Control/9.03%3A_PID_Tuning_via_Classical_Methods?utm_source=)

de Winkel, K.N., Irmak, T., Happee, R. and Shyrokau, B. (2023) 'Standards for passenger comfort in automated vehicles: Acceleration and jerk', *Applied Ergonomics*, 106, pp. 103881. Available at: <https://doi.org/10.1016/j.apergo.2022.103881>.

Fatah, I.S. (2014) 'PSO-based tuning of PID controller for speed control of DC motor', *Diyala Journal of Engineering Sciences*, 7(3), pp. 65–79. Available at: <https://doi.org/10.24237/djes.2014.07304>.

Halder, K., Das, S. and Gupta, A. (2021) 'Time delay handling in dominant pole placement with PID controllers to obtain stability regions using random sampling', *International Journal of Control*, 94(12), pp. 3384–3405. Available at: <https://doi.org/10.1080/00207179.2020.1764110>.

International Organization for Standardization (2022) *ISO 21448:2022(en)*.

International Organization for Standardization (2018a) *ISO 15622:2018(en)*.

International Organization for Standardization (2018b) *ISO 26262-1:2018(en)*.

Iskandar, E., rameli, m. and Fatoni, A. (2019) 'Artificial neural network approach for parameter estimation in PI self tuning regulator (PI-STR) method on process rig 38-714 pressure control', *JAREE (Journal on Advanced Research in Electrical Engineering) (Online)*, 3(1).

Mitov, A., Slavov, T., Krlev, J. and Angelov, I. (2023) 'Axial-piston pump optimal multivariable PID controller tuning', *IOP Conference Series.Earth and Environmental Science*, 1234(1), pp. 12013. Available at: <https://doi.org/10.1088/1755-1315/1234/1/012013>.

Om Bansal, H., Sharma, R. and Shreeraman, P.R. (2012) 'PID controller tuning techniques: A review', 2(4), pp. 68–176. Available at:

https://www.researchgate.net/publication/316990192_PID_Controller_Tuning_Techniques_A_Review.

Patel, V.V. (2020) 'Ziegler-nichols tuning method: Understanding the PID controller', *Resonance*, 25(10), pp. 1385–1397. Available at: <https://doi.org/10.1007/s12045-020-1058-z>.

Sung, S.W., Lee, I. and Lee, B. (1998) 'On-line process identification and automatic tuning method for PID controllers', *Chemical Engineering Science*, 53(10), pp. 1847–1859. Available at: [https://doi.org/10.1016/S0009-2509\(98\)00012-8](https://doi.org/10.1016/S0009-2509(98)00012-8).

Tandan, N. and Kumar Swarnkar, K. (2015) 'PID controller optimization by soft computing techniques-A review', *International Journal of Hybrid Information Technology*, 8(7), pp. 357–362. Available at: https://www.researchgate.net/publication/283739427_PID_Controller_Optimization_By_Soft_Computing_Techniques-A_Review.

Turan, A. (2025) 'Improved PID control design for electric power steering DC motor', *IEEE Access*, 13, pp. 6080–6088. Available at: <https://doi.org/10.1109/ACCESS.2024.3524303>.