

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

Energy Storage and Control Systems (ACHA7007)

Taylor Anais Rohn

2500151

Date 29/03/2026

Contents

1. Methodology & Workflow 3

1.1 Encoding and Decoding..... 3

1.2 Controller feedback 4

1.3 Number of bits 11

1.4 Sampling time..... 15

1.5 Delay 16

1.6 ISO 15622 regulations 18

2 Conclusion..... 19

3. References 20

1. Methodology & Workflow

1.1 Encoding and Decoding

To be able to use the velocity sensor reading for the controller, it is easier to decode the binary value of velocity raw into a base 10 physical value Velocity.

By following the Eq. (1.1) the result is Eq (1.2).

$$X_{Physical} = \frac{X_{Raw} \times (M - m)}{R} + m \quad (1.1)$$

$$Velocity = \frac{Velocity_{Raw} \times (M_{velocity_S} - m_{velocity_S})}{R} + m_{velocity_S} \quad (1.2)$$

The velocity demand signal is obtained from the driver by the function getRawVelocityDemand. Once again, this is a 16-bit encoded binary signal which needed to be decoded to be used easily within the controller. By following Eq (1.1) the result is (1.3).

$$Demand = \frac{Demand_{Raw} \times (M_{demand} - m_{demand})}{R} + m_{demand} \quad (1.3)$$

The control signal is to be encoded into a 16-bit binary number to be sent into the emulated communication bus. You need to write the function for the encoding equation is Eq (1.4) and resulting equation is Eq (1.5).

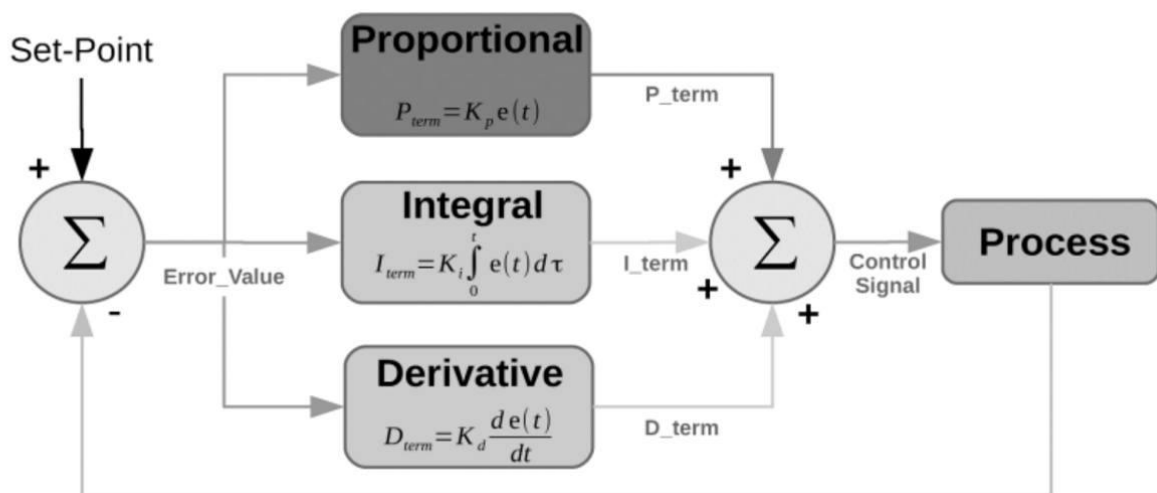
$$X_{Raw} = \frac{(X_{Physical} - m) \times R}{M - m} \quad (1.4)$$

$$Control_{Raw} = \frac{(Control - m_{ControlSig}) \times R}{(M_{ControlSig} - m_{ControlSig})} \quad (1.5)$$

1.2 Controller feedback

The controller takes as input the velocity, the previous control signal, the error, the demand, the simulation time step, the relevant gains, the sampling time and a counter and delivers as output the control signal and error.

The PID controller is described in Fig. 1.2.1:



[Picture 1.2.1: PID Controller diagram]

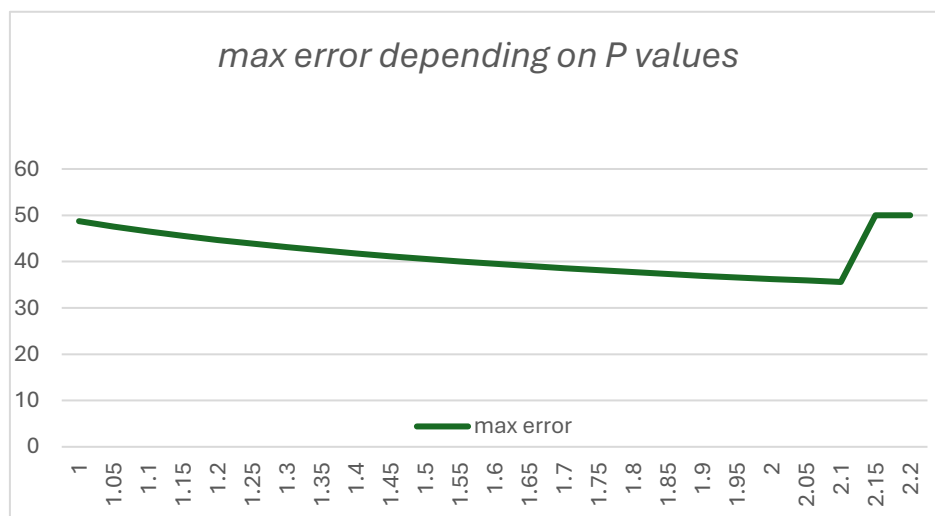
By creating the persistent variable `prev_I_error` local to the function in which it was declared, its value is retained in memory between calls to the function. This represents an easy way to build the integral gain. The remaining gains are simply implemented in the function by following their defining equations.

When investigating the trade-offs between velocity response and control effort it must be noted that the diminishing of the control effort of the control output is possible and leads to minimising energy usage and reduced wear of the control system, however, may significantly worsen tracking properties of the control system. This applies to settling times and overshoots as well. [1].

When selecting the proportional gain, the main goal was to find the highest value that would speed up the response of the PID without putting the process into oscillation. [2]

P	I	D	max error	average error
1	0	0	48.7025	31.4175
1.05	0	0	47.549	30.3158
1.1	0	0	46.5053	29.3013
1.15	0	0	45.5463	28.364
1.2	0	0	44.6696	27.4938
1.25	0	0	43.8571	26.6829
1.3	0	0	43.1064	25.9245
1.35	0	0	42.406	25.214
1.4	0	0	41.7491	24.545
1.45	0	0	41.1357	23.9147
1.5	0	0	40.5612	23.3192
1.55	0	0	40.0188	22.7559
1.6	0	0	39.5061	22.2213
1.65	0	0	39.0208	21.7132
1.7	0	0	38.5631	21.2299
1.75	0	0	38.1282	20.769
1.8	0	0	37.7139	20.3299
1.85	0	0	37.318	19.91
1.9	0	0	36.9426	19.5082
1.95	0	0	36.581	19.1238
2	0	0	36.2399	18.7547
2.05	0	0	35.9103	18.4006
2.1	0	0	35.5945	18.1013
2.15	0	0	50	17.8856
2.2	0	0	50	17.5695

[Table 1.2.1: selection of P gain]



[Graph 1.2.1: max error depending on P values]

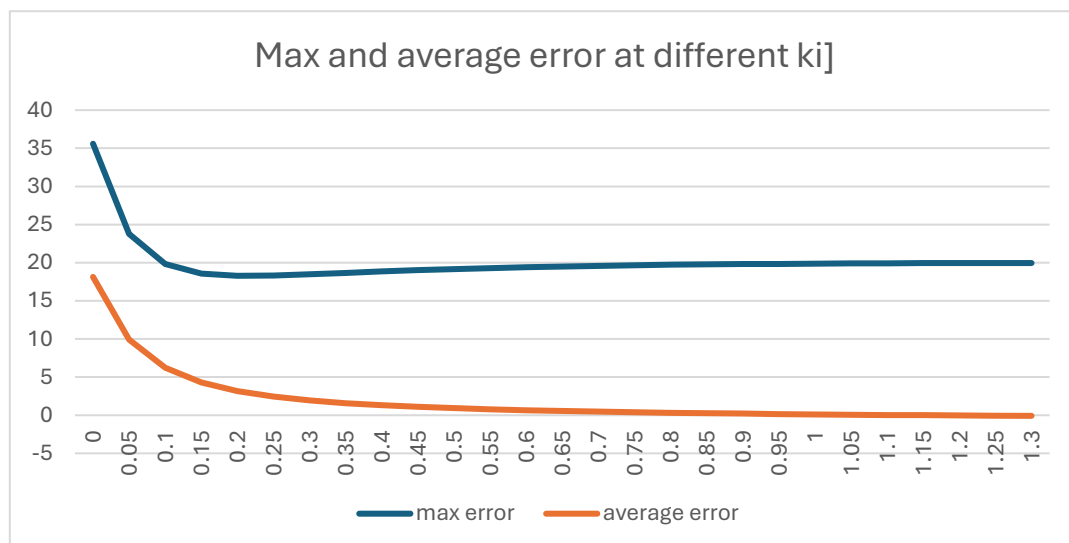
The integral gain is directly related to performance and the proportional gain directly related to robustness. [2]

When selecting I the main aspect was to reduce both the steady state error to zero while avoiding the *integral windup* [3], which can occur when the value of the integral saturates the PID controller and the error signal cannot be driven to zero, and the aggressive initial overshoot.

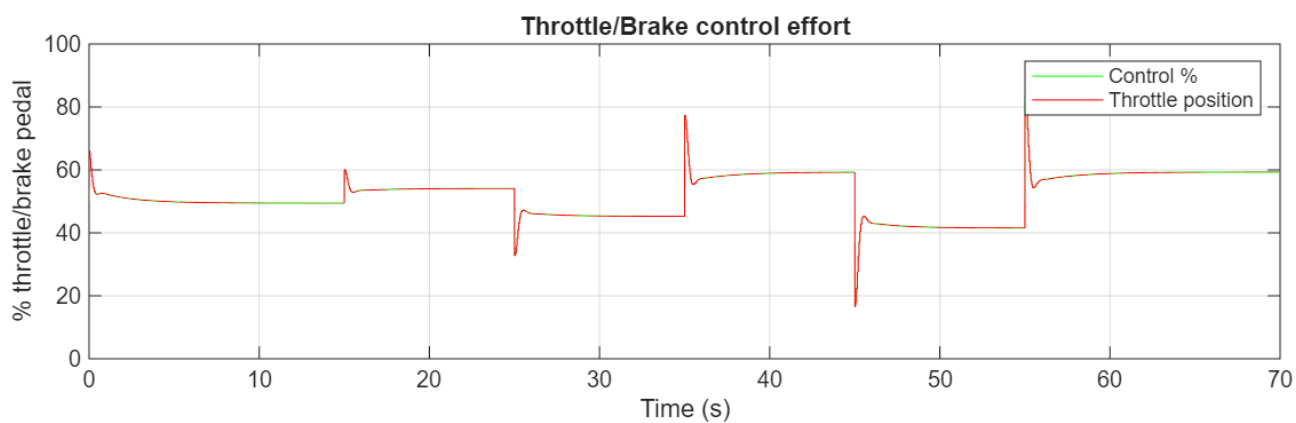
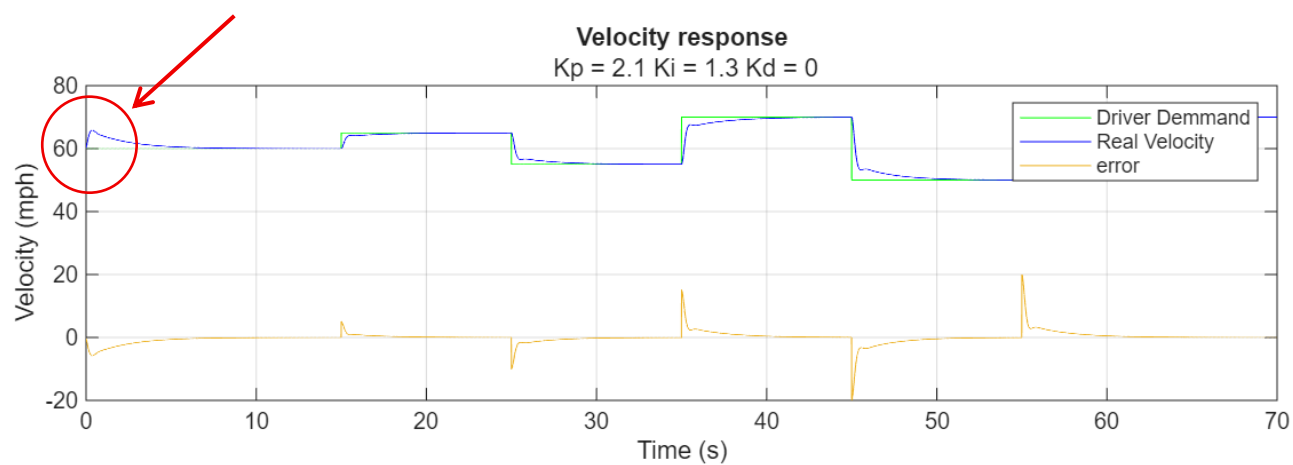
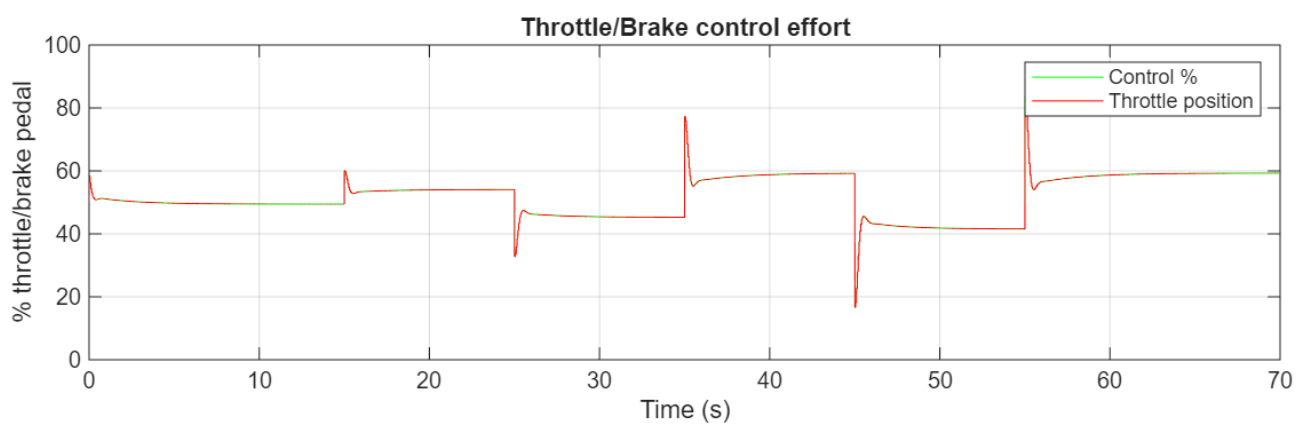
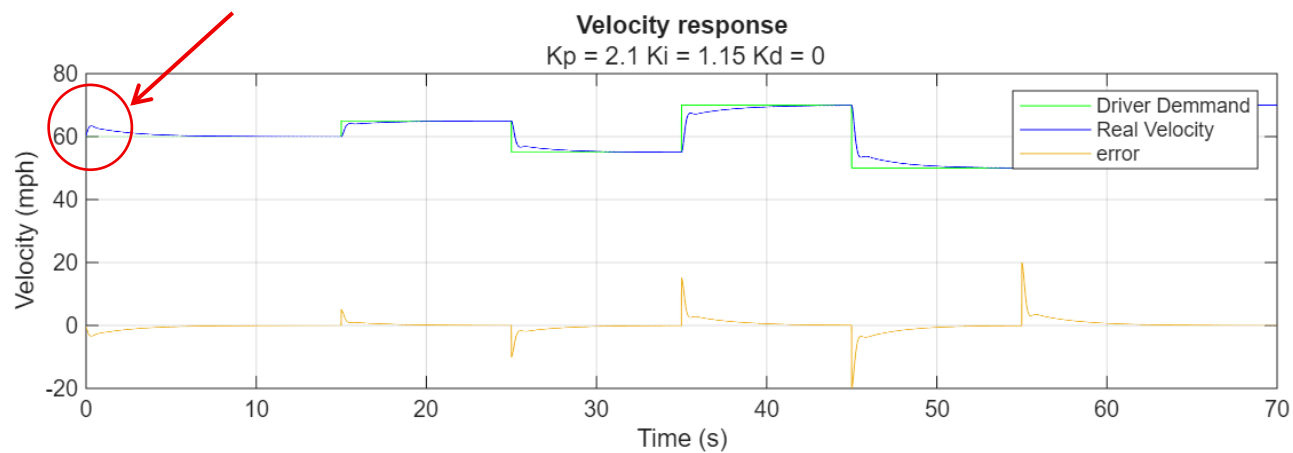
As observed in Table 2.2 in the average error column, the mentioned overshoot gets increasingly worse for $k_i > 1.15$. In real life scenarios an excessive initial overshoot would translate in the car dangerously accelerating when the cruise control gets set.

P	I	D	max error	average error
2.1	0	0	35.5945	18.1345
2.1	0.05	0	23.7567	9.9316
2.1	0.1	0	19.8405	6.2291
2.1	0.15	0	18.5817	4.2911
2.1	0.2	0	18.2796	3.1616
2.1	0.25	0	18.3208	2.4415
2.1	0.3	0	18.4787	1.9475
2.1	0.35	0	18.6664	1.5885
2.1	0.4	0	18.8495	1.3154
2.1	0.45	0	19.0189	1.1004
2.1	0.5	0	19.1699	0.9263
2.1	0.55	0	19.3004	0.7823
2.1	0.6	0	19.4125	0.6611
2.1	0.65	0	19.511	0.5577
2.1	0.7	0	19.5934	0.4683
2.1	0.75	0	19.6643	0.3903
2.1	0.8	0	19.7238	0.3216
2.1	0.85	0	19.7719	0.2608
2.1	0.9	0	19.8131	0.2064
2.1	0.95	0	19.8474	0.1576
2.1	1	0	19.8749	0.1136
2.1	1.05	0	19.8978	0.0736
2.1	1.1	0	19.9184	0.0373
2.1	1.15	0	19.9344	0.004
2.1	1.2	0	19.9458	-0.0266
2.1	1.25	0	19.9573	-0.0547
2.1	1.3	0	19.9664	-0.0807

[Table 1.2.2: Selection of I gain]

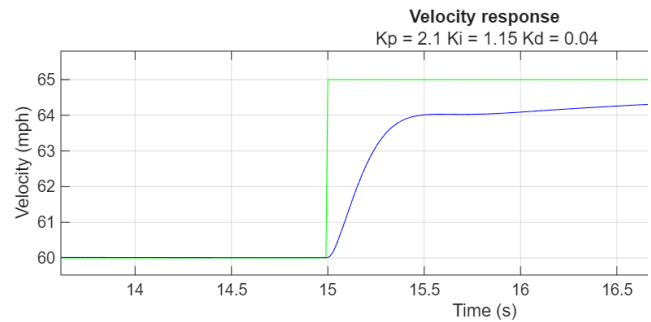
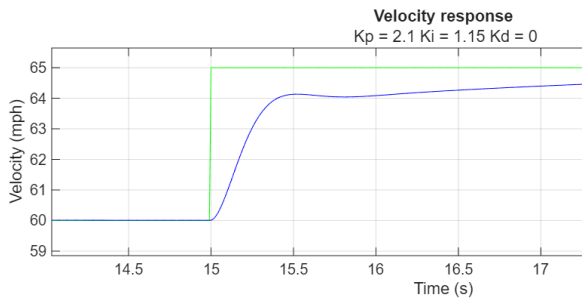


[Graph 1.2.2: Max and average error at different k_i]

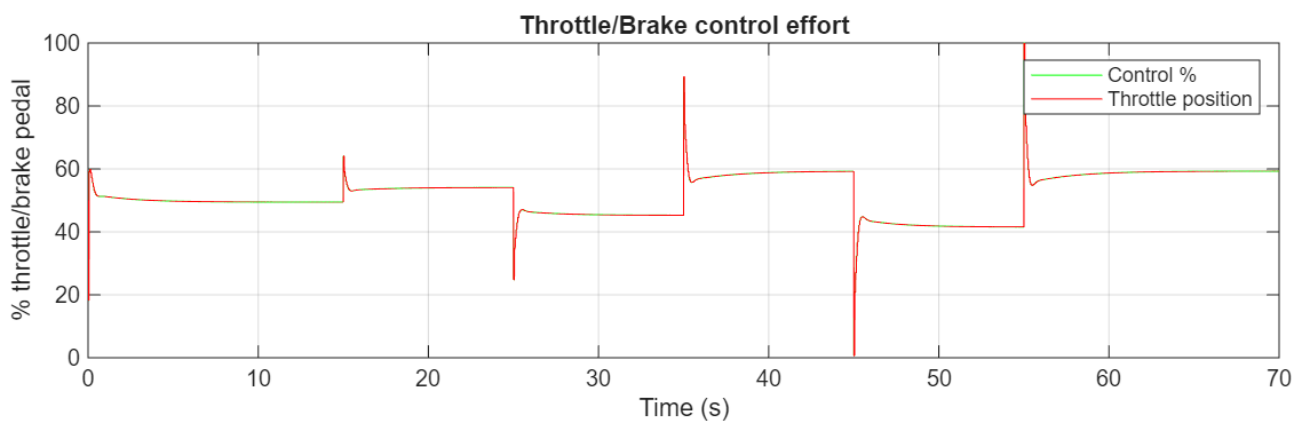
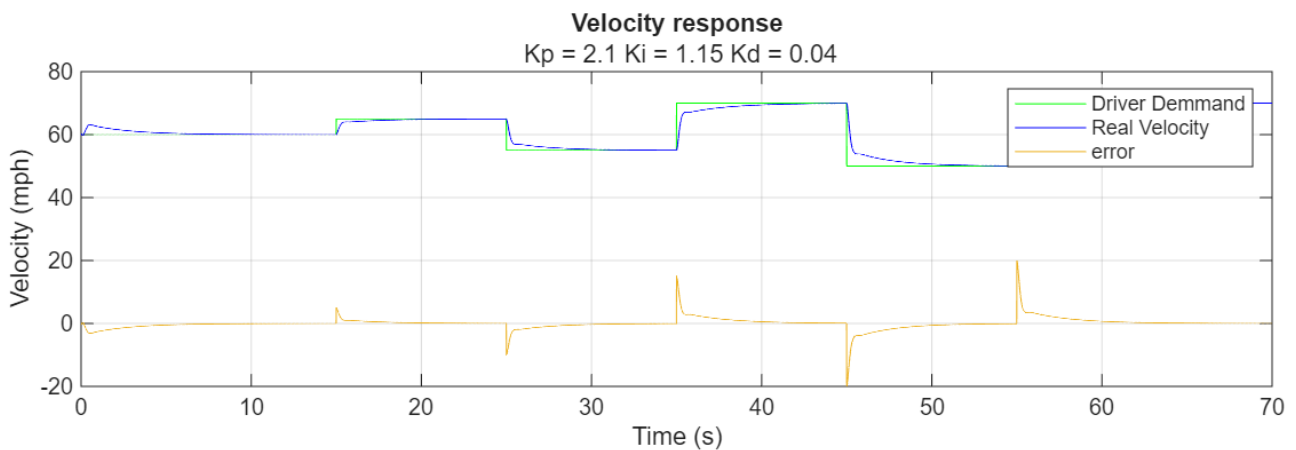


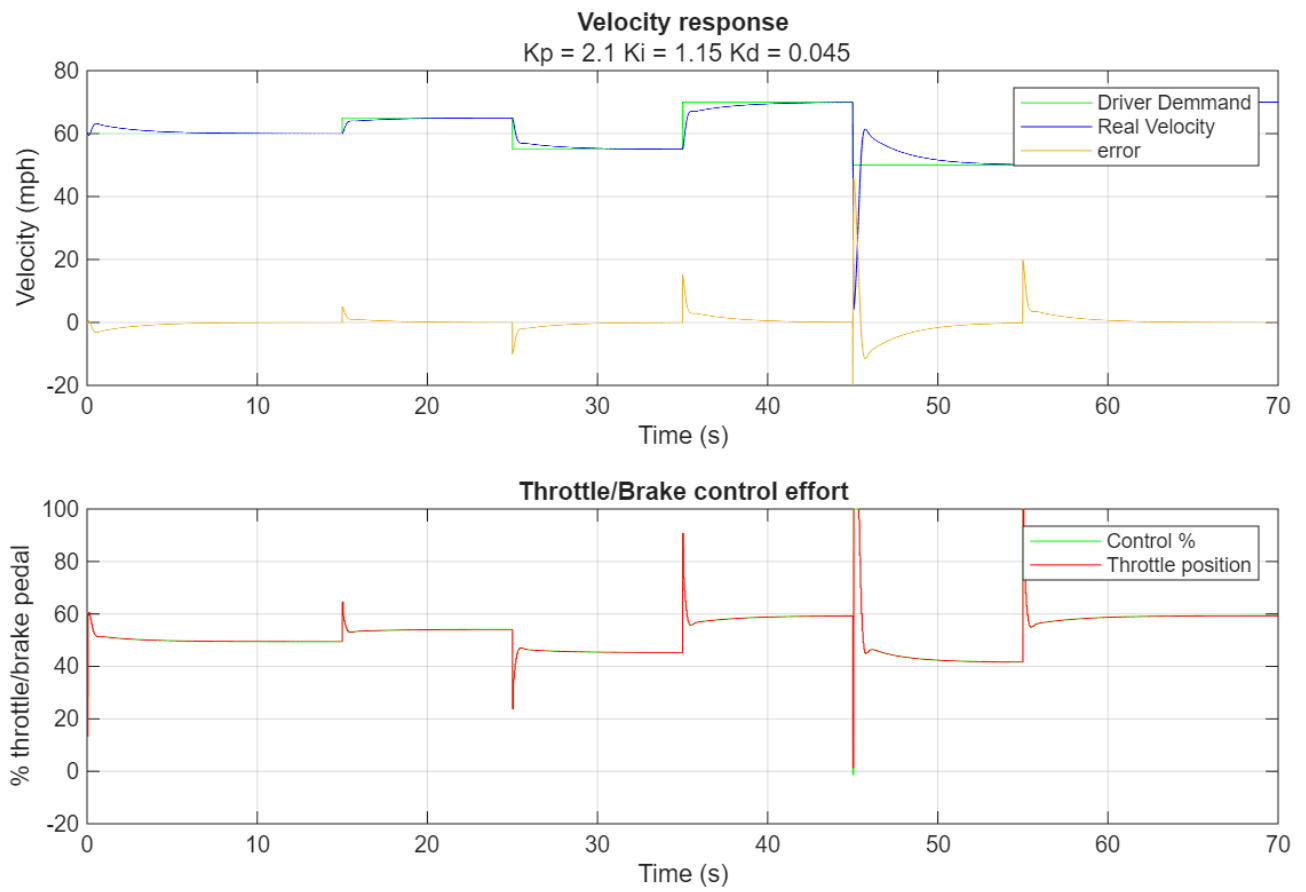
[Graph 1.2.3: initial overshoots highlighted at different k_i values]

The *derivative* component was chosen looking at the rate of change in the output. Increasing this value will cause the output to react in very small increments; if increments are too small, the derivative can cause the loop to become sensitive to noise [3]. In Graph 2.4 can be noted smoother progression without sudden overshoots which instead occur in Graph 2.5.



[Graph 1.2.4: velocity response at $k_d = 0$ (a) and $k_d = 0.4$ (b)]

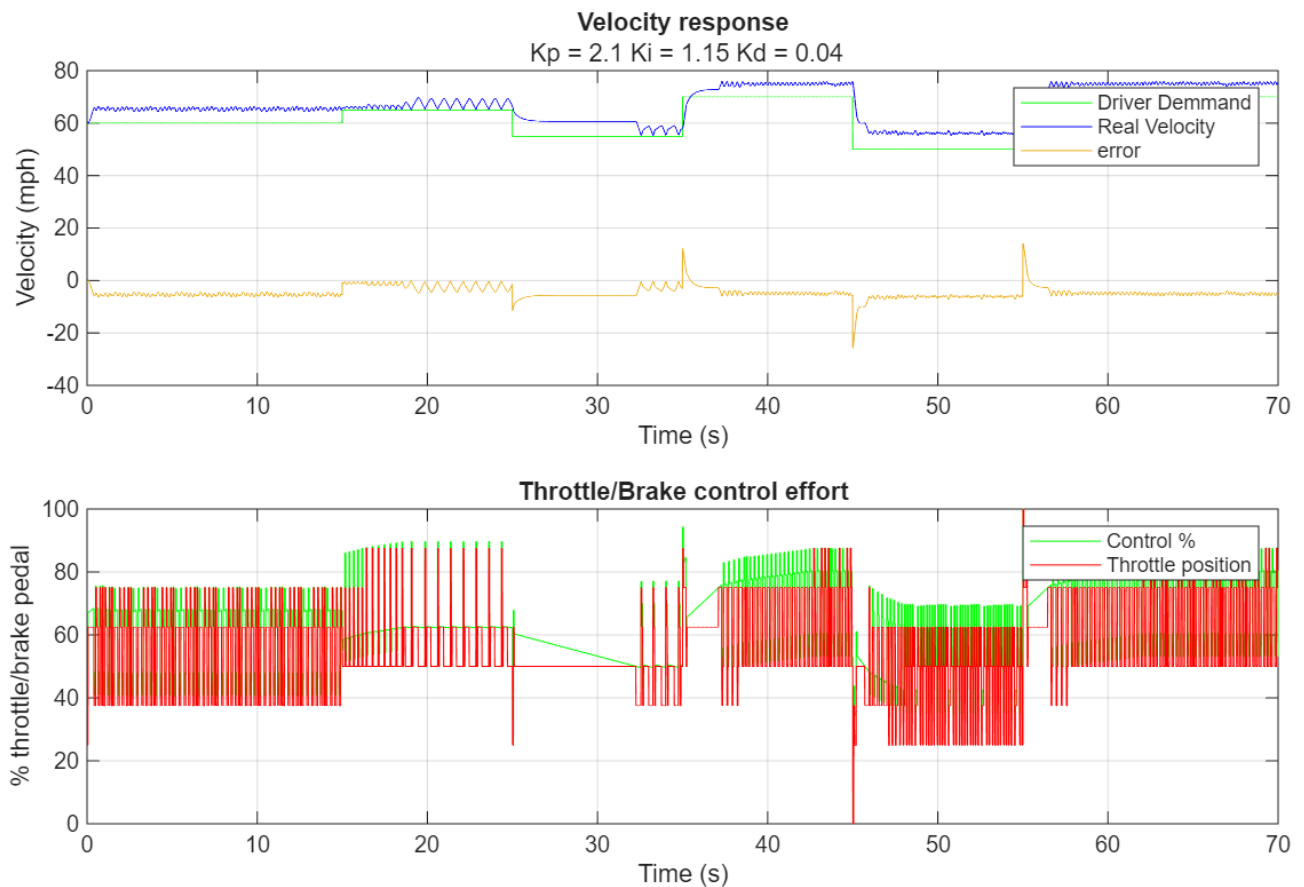




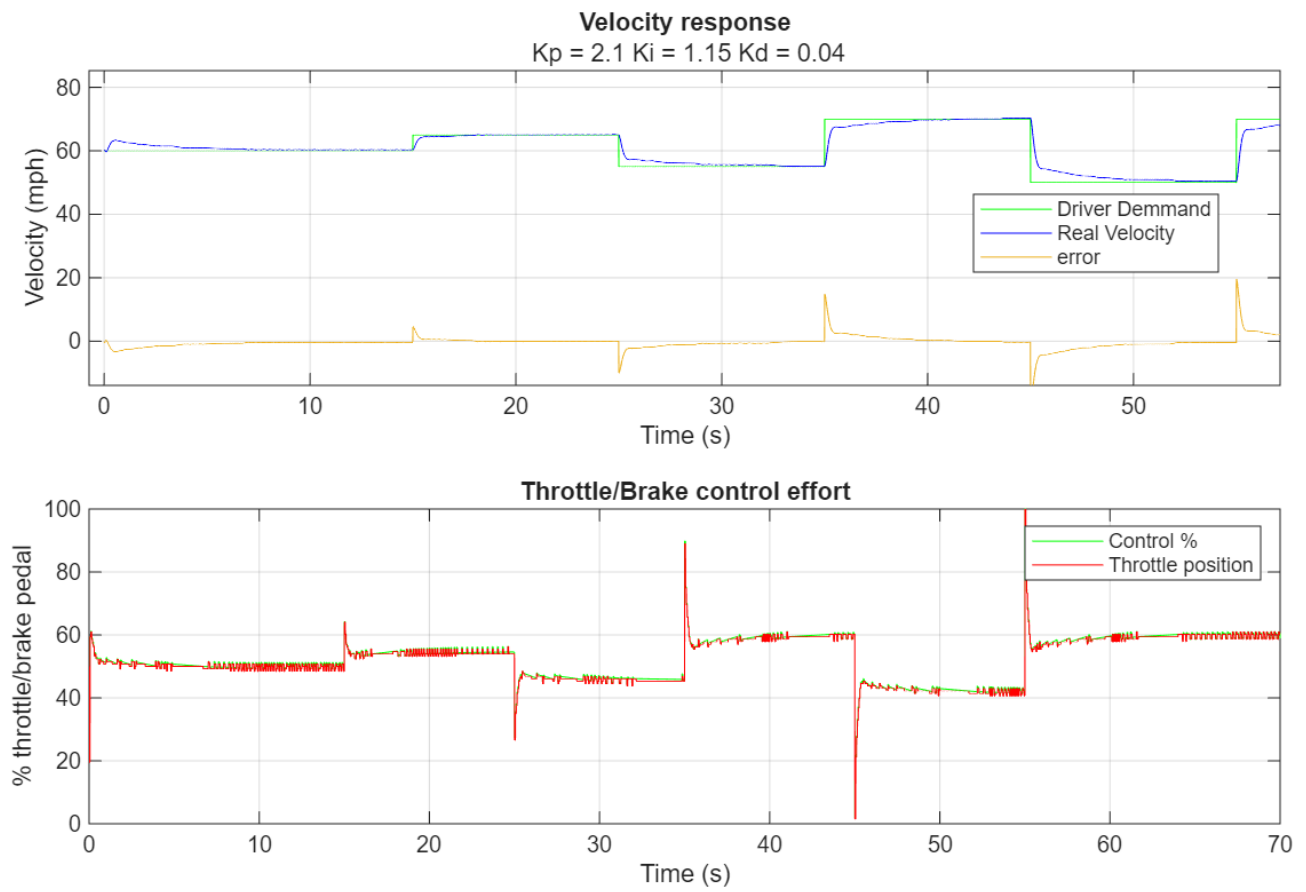
[Graph 1.2.5: velocity response at $k_d = 0.04$ (a) and $k_d = 0.45$ (b)]

1.3 Number of bits

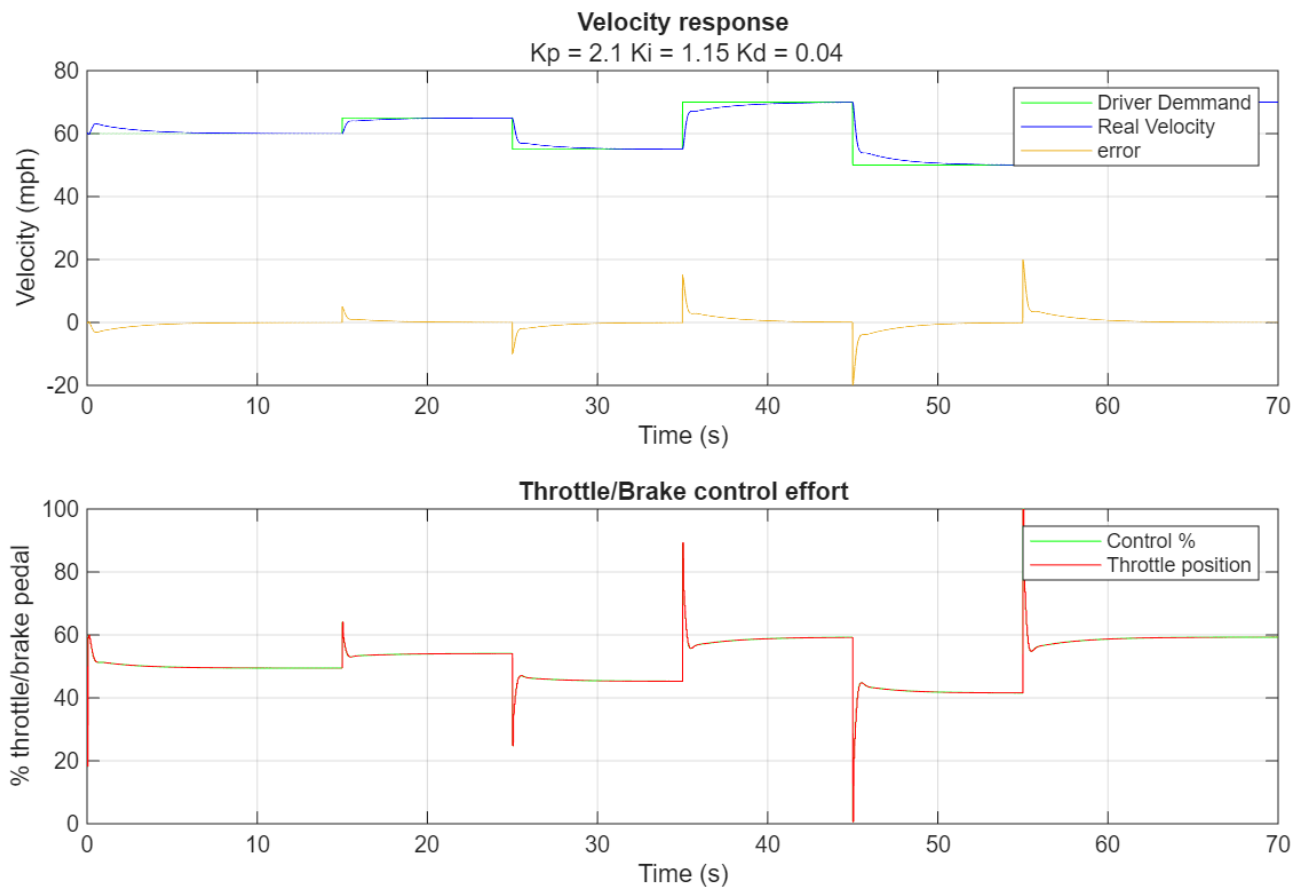
Less bits translate in less information to be read, stored and transmitted, resulting in faster transmission times, less used storage, but as a downside it means less details and more disturbance. In this case at 4 bits the data transmitted is unrecognisable, at 8 bits still guarantees reasonable performance, at 16 bits is optimal and at 32 is unnecessarily overarching.



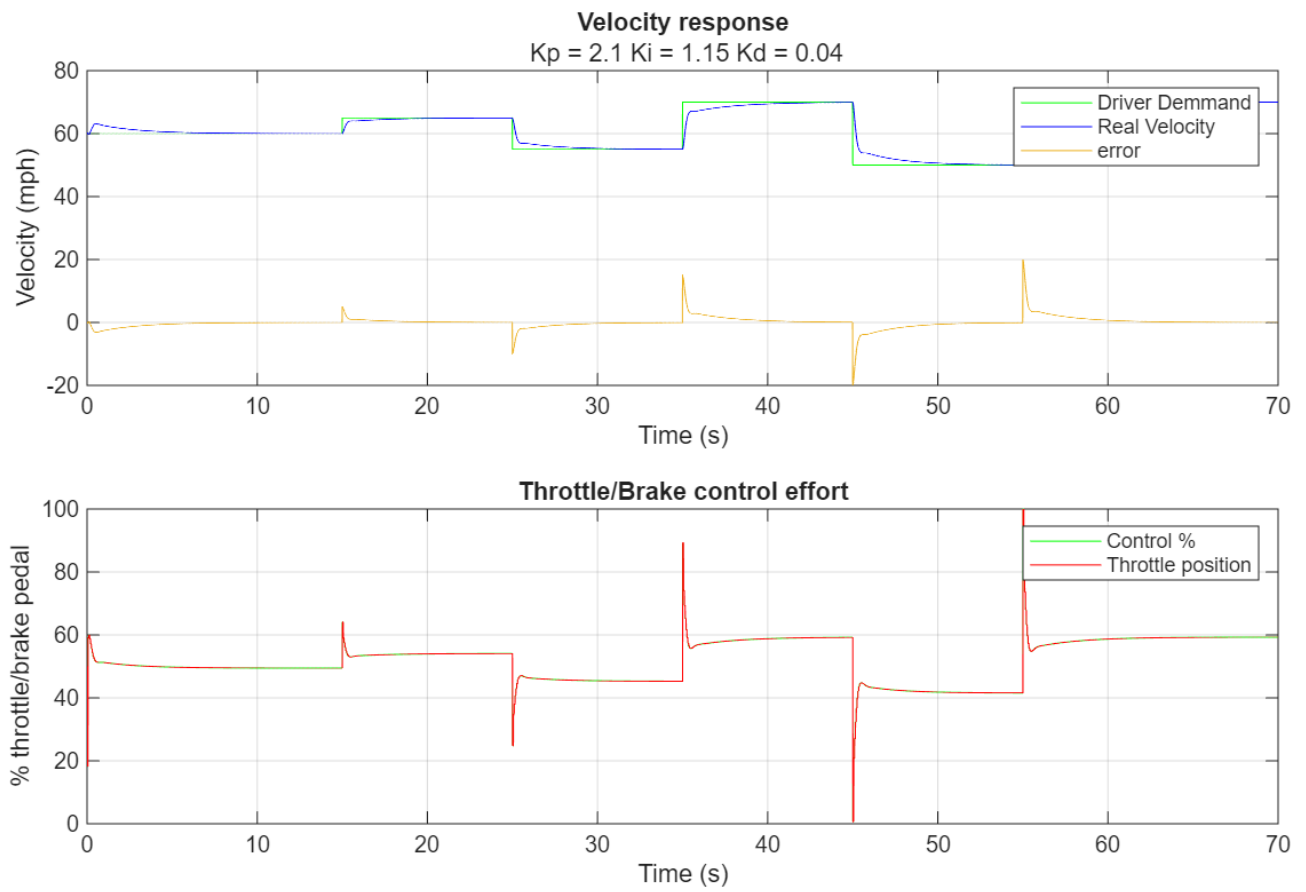
[Graph 1.3.1: velocity response and throttle/brake control effort at 4 bits]



[Graph 1.3.2: velocity response and throttle/brake control effort at 8 bits]



[Graph 1.3.2: velocity response and throttle/brake control effort at 16 bits]



[Graph 1.3.1: velocity response and throttle/brake control effort at 32 bits]

1.4 Sampling time

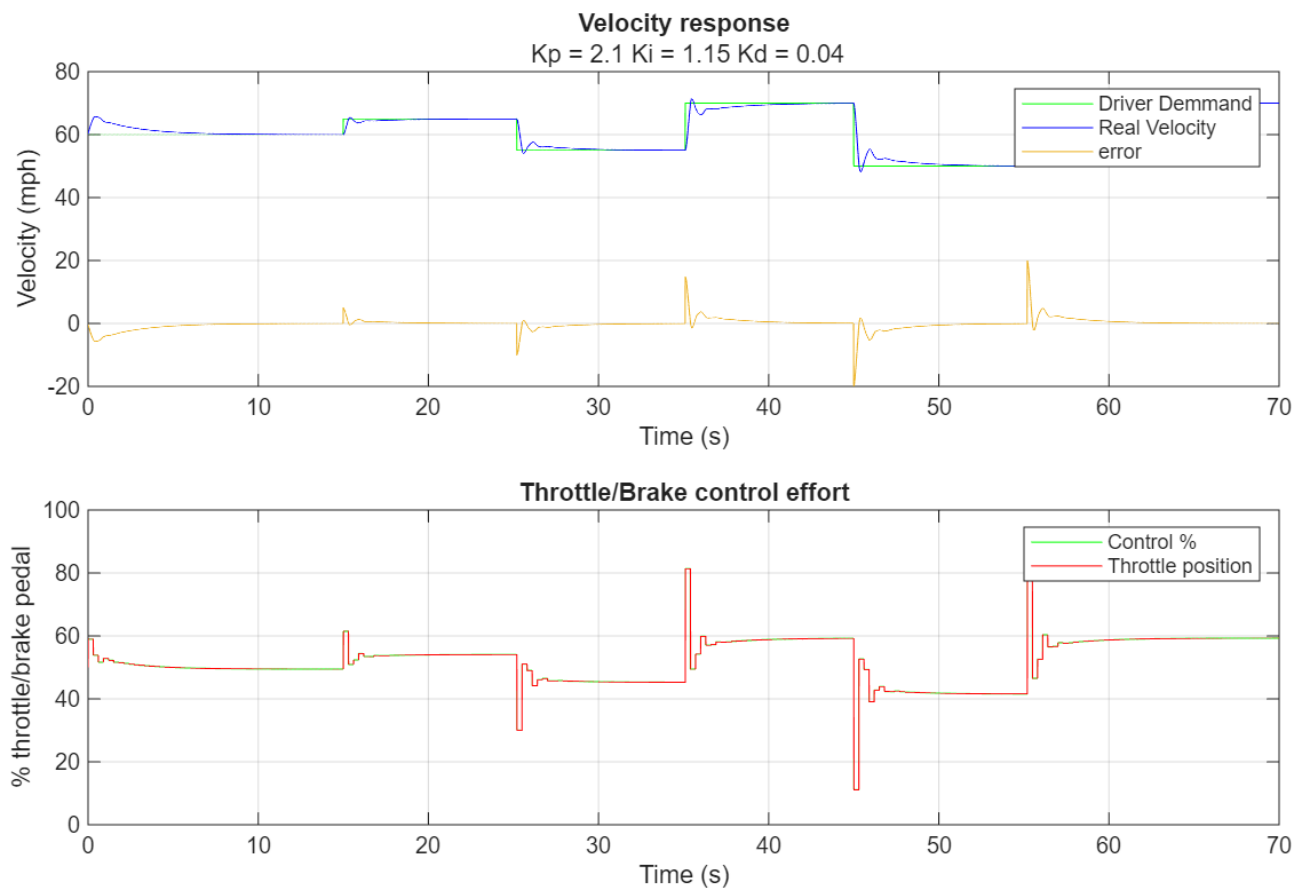
Lower sampling times result in longer periods of time between samples, less data collected but faster data processing. A too long sampling time has several consequences, including Aliasing, loss of information with consequential loss of ability to replicate the input wave form. In this case a longer sampling time means a longer time required to reach the setpoint after every change in the input. [5]

Sampling times in real case scenarios usually follow the Nyquist–Shannon sampling theorem for periodic signals, Eq (1.3.1)

$$F_s \geq 2f_{max} \quad (1.3.1)$$

For non-periodic signals it is usually best to use the max frequency registered and still apply the Nyquist-Shannon sampling theorem, or use an adaptive method.

In this case the longest sampling time that guarantees reasonable performance is $T_s = 0.3$ s.



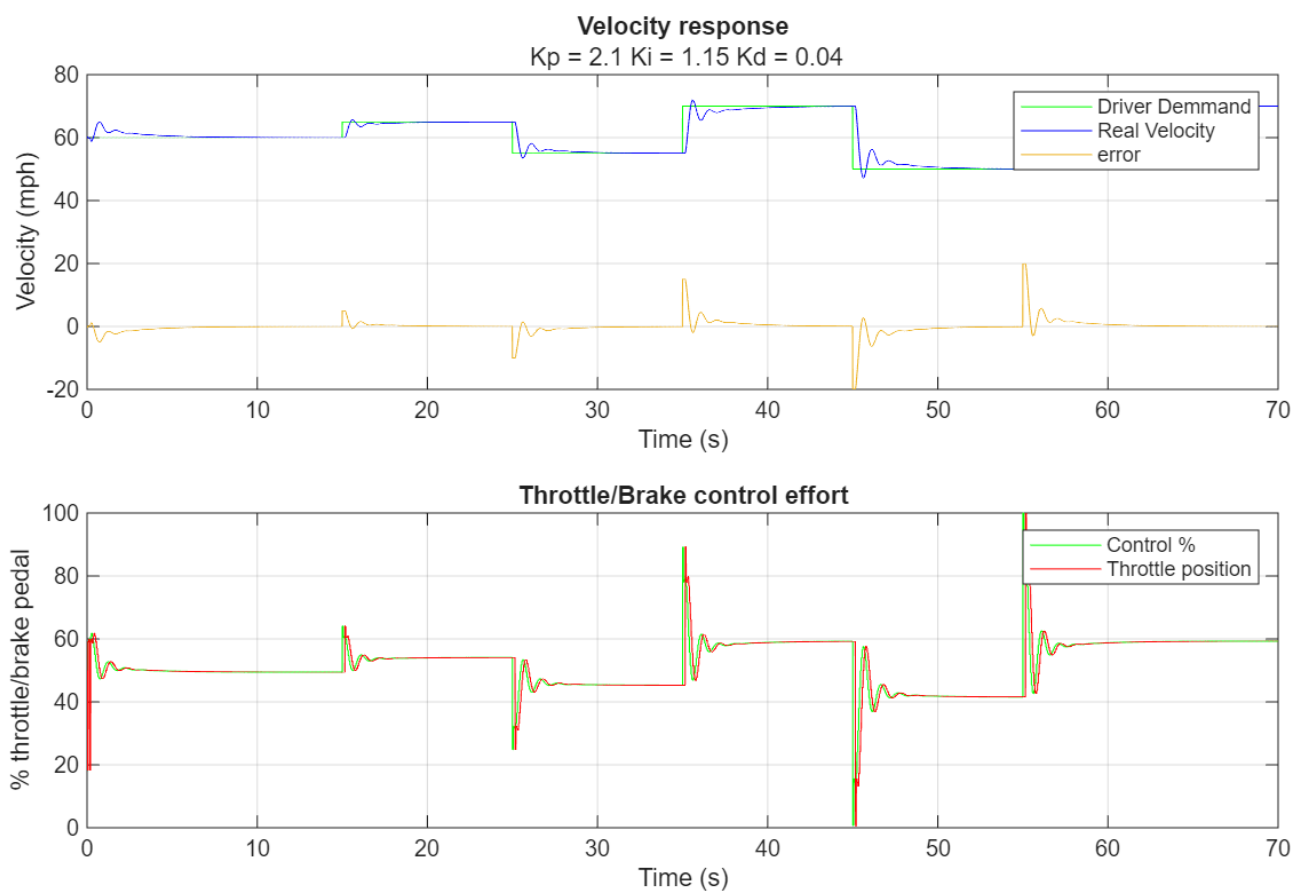
[Graph 1.4.1: velocity response and Throttle/brake control effort at $T_s = 0.3$ s]

1.5 Delay

Network delays negatively affect communication by delaying response to stimuli, causing buffering in reaching the setpoint. When the delay is too high the system might not be able to counteract changes in the signal, over elongating signals might get over stacked.

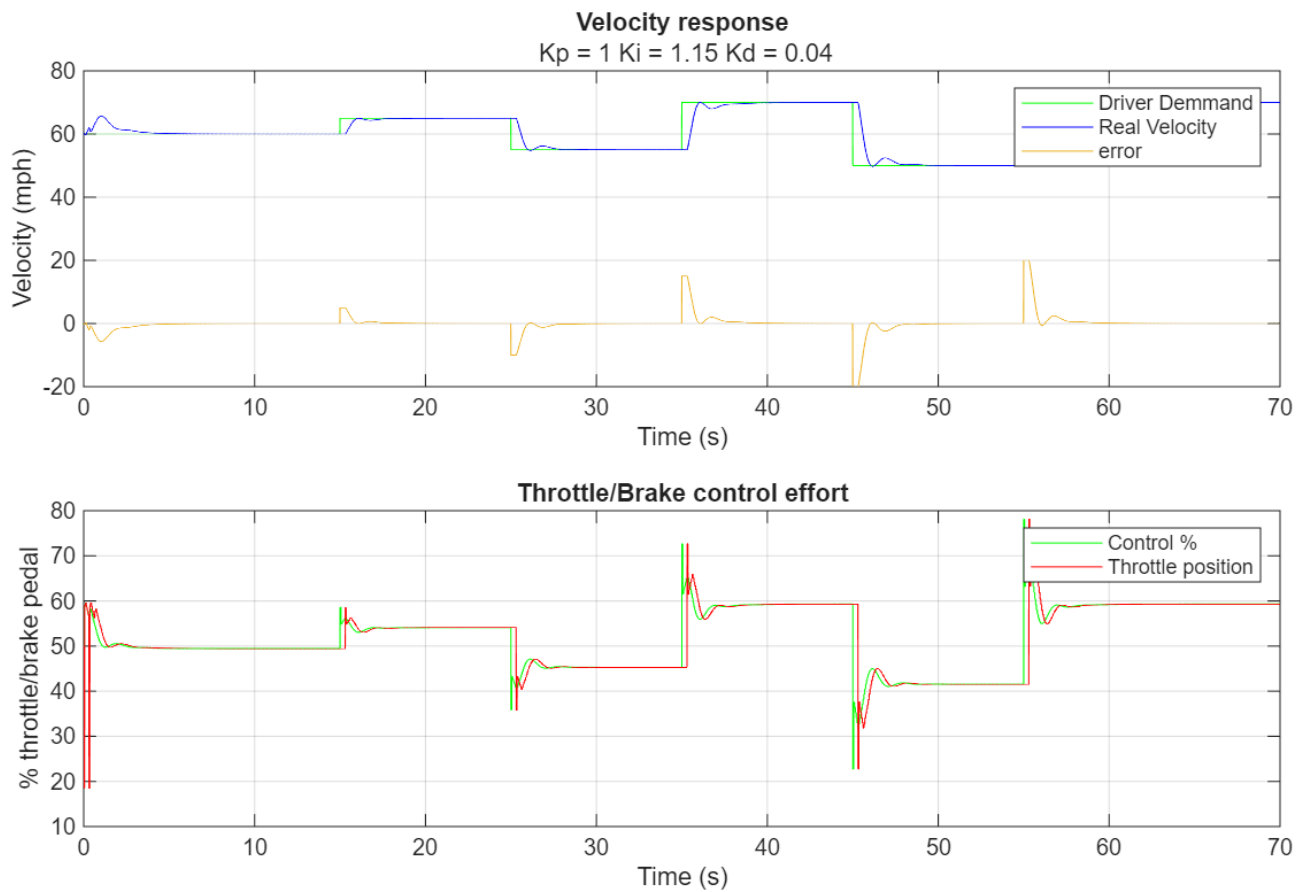
At maximum delay = 0.15 s a satisfactory performance can still be achieved.

a

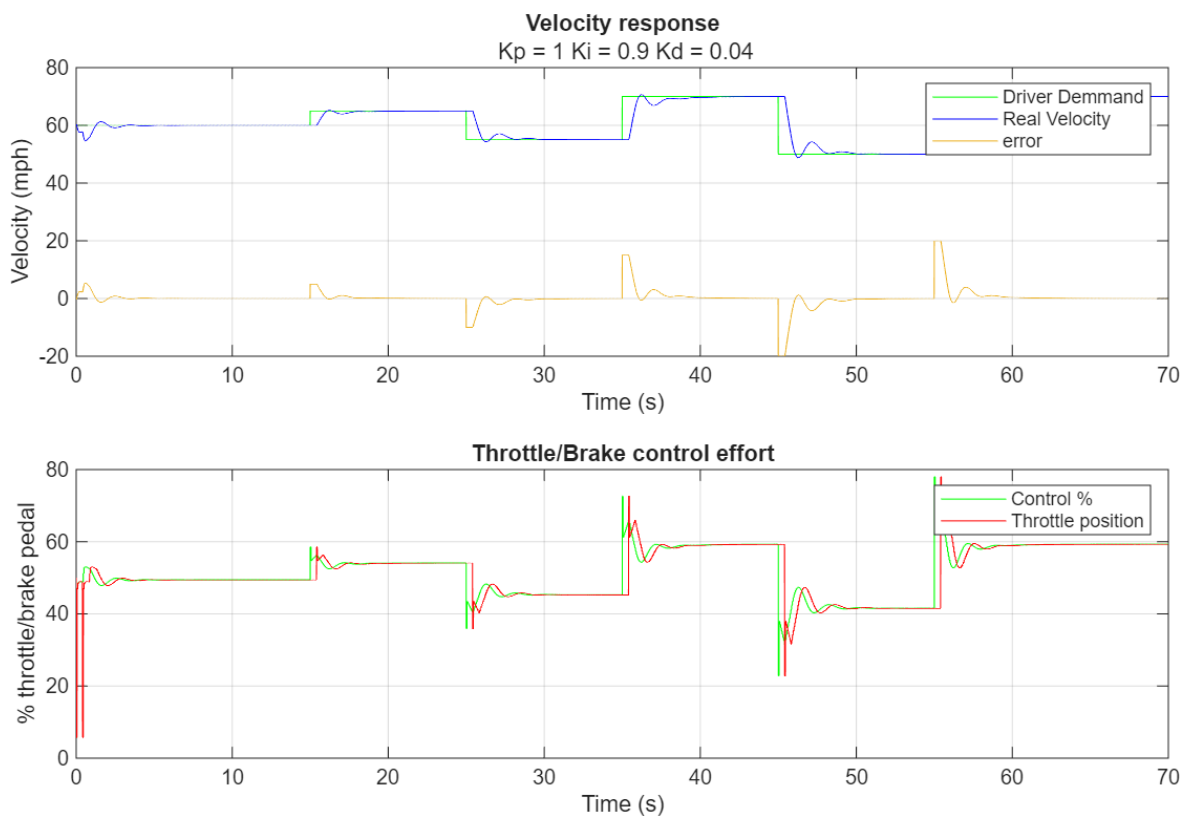


[Graph 1.5.1: Velocity response and Throttle/Brake control effect at delay = 0.15 s]

By lowering P and I gains the system can withstand a longer delay while delivering a satisfactory performance. [cite document 1]



[Graph 1.5.2: Velocity response and Throttle/Brake control effect at delay = 0.3 s]



[Graph 1.5.4: Velocity response and Throttle/Brake control effect at delay = 0.4s]

1.6 ISO 15622 regulations

ISO states that “the minimum selectable time gap ($\tau_{\min}$) shall be the minimum selectable time gap for following control mode under steady-state conditions for all speeds v .

$\tau_{\min}$ shall be greater than or equal to 0,8 s.

At least one time gap setting the time gap between vehicles (τ) in the range of 1,5 s to 2,2 s shall be provided for speeds higher than 8 m/s.

Under steady-state conditions the minimum clearance shall be $MAX(c_{\min}, \tau_{\min} \cdot v)$.

Under transient conditions, the clearance may temporarily fall below the minimum clearance. If such a situation occurs, the system shall adjust the clearance to attain the desired clearance.

For all Full speed range adaptive cruise control type systems (FSRA) and those Limited Speed Range Adaptive Cruises Control (LSRA) type systems which attempt to stop behind a stopping vehicle, as a minimum requirement the system shall be able, starting from steady state following, to stop behind a gradually stopping vehicle which is braking with the acceleration a_{stopping} at a speed below $v_{\text{stopping}} = 10 \text{ m/s}$.” [6]

The PID Controller described in this document complies with the ISO regulations.

2 Conclusion

The PID Controller here presented and discussed has reasonable P, I, D values that achieve reasonable performance and show fair robustness to delays and complying to ISO

3. References

1	Bartyś, M., & Hryniewicki, B. (2019). The Trade-Off between the Controller Effort and Control Quality on Example of an Electro-Pneumatic Final Control Element. <i>Actuators</i> , 8(1), 23. https://doi.org/10.3390/act8010023
2	Olof Garpinger, Tore Hägglund, Karl Johan Åström, (2014) Performance and robustness trade-offs in PID control, <i>Journal of Process Control</i> , Volume 24, Issue 5, Pages 568-577, ISSN 0959-1524,
3	Tuning a PID Control Loop Technical Note Form 2171-200731
4	V. Romero Segovia, T. Hägglund, K.J. Åström, (2014), Measurement noise filtering for PID controllers, <i>Journal of Process Control</i> , Volume 24, Issue 4 Pages 299-313, ISSN 0959-1524, https://doi.org/10.1016/j.jprocont.2014.01.017
5	E. Sefer, M. Kleyman, Z. Bar-Joseph, (2016), Tradeoffs between Dense and Replicate Sampling Strategies for High-Throughput Time Series Experiments <i>Cell Systems</i> , Volume 3, Issue 1, Pages 35-42, ISSN 2405-4712,
6	ISO 15622:2018, Intelligent transport systems — Adaptive cruise control systems — Performance requirements and test procedures