

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)

Iván Moreno Fernández

2504057

Date (03/2026)

1. Encoding and decoding function for the sensor, demand and control signal.

Using the bin2dec function from MATLAB (*Mathworks.com (2019)*), it is possible to convert either a binary number to decimal number or vice versa. In addition to the function, a mathematical operation is performed to put the number that is obtained in the range of values the velocity cruise control works with.

Equation 2. From raw binary to decimal

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

Equation 1. From decimal to raw binary

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

2. Code developed for the feedback controller and its coefficients.

The code developed for the feedback controller was based on a PID type control. The function will give two outputs, the control signal and the error between the real velocity and the driver's demand. The inputs that the function requires are velocity, demand, previous signal control, PID coefficients, sampling time, time step, the counter from the main loop and the error's array.

At first, it is necessary to make some control statements for the correct performance of the PID. With the persistent function (*Mathworks.com (2026)*) it can be saved the value of previous integral error so it can be accumulated along the loop. In addition, a conditional loop is developed to ensure an initial value for previous error in the first iterations.

After these previous establishment, the main code of the function is developed, for each part of a PID control.

- Proportional part: multiplying the error by the proportional coefficient
- Integral part: As it is calculate considering the previous error for a better adjustment. It is important to remark updating the value of the integral part for the next iterations of the loop.
- Derivate part: Based on the difference between the actual error and the previous one during the time sample.

All the three parts are combined in the control signal will be sent to change the throttle position. Because of the throttle limitation, a statement is created in the

function to limit the value of the control signal therefore cannot be higher than the limits of the throttle position.

The Coefficient values were selected based on the better performance, which means the ones approach a 0 steady state error. To achieve this configuration, a manual iterative process was developed based on different iterative processes explained in technical papers (*Chen, X. et al 2023; Jiang, S. et al 2009*). Starting with the proportional value, it was increased in 1 by 1 steps until reach an oscillatory behaviour around the demand (4 in this case); from this state, the proportional value will be fixed in a range of 40-60% of this initial value (1.8 for this case), and the iterative change starts with the integral coefficient, from 0 to 0.8 in steps of 0.1, to eliminate the steady state error attenuating the response. Regarding the derivative coefficient, the iterative process began from 0 to 0.05 with steps of 0.01 until the overshoot from the signal is controlled and has a smooth response.

The priority in this case was to obtain a smoother response from the control, therefore can build up consistently until reach the demand, rather than having a high velocity response that could lead in high peaks of acceleration that could be uncomfortable for the driver.

3. Sensitivity analysis of the number of bits for signal encoding/decoding

After doing a study from a 4 bits signal to 64 for bits, the limits for the encoding and decoding will be 4 bits as the minimum value to obtain a reasonable response, and 32 as the maximum value in the current configuration, as how the code is written, it does not have the capacity to storage numbers higher than 32 bits.

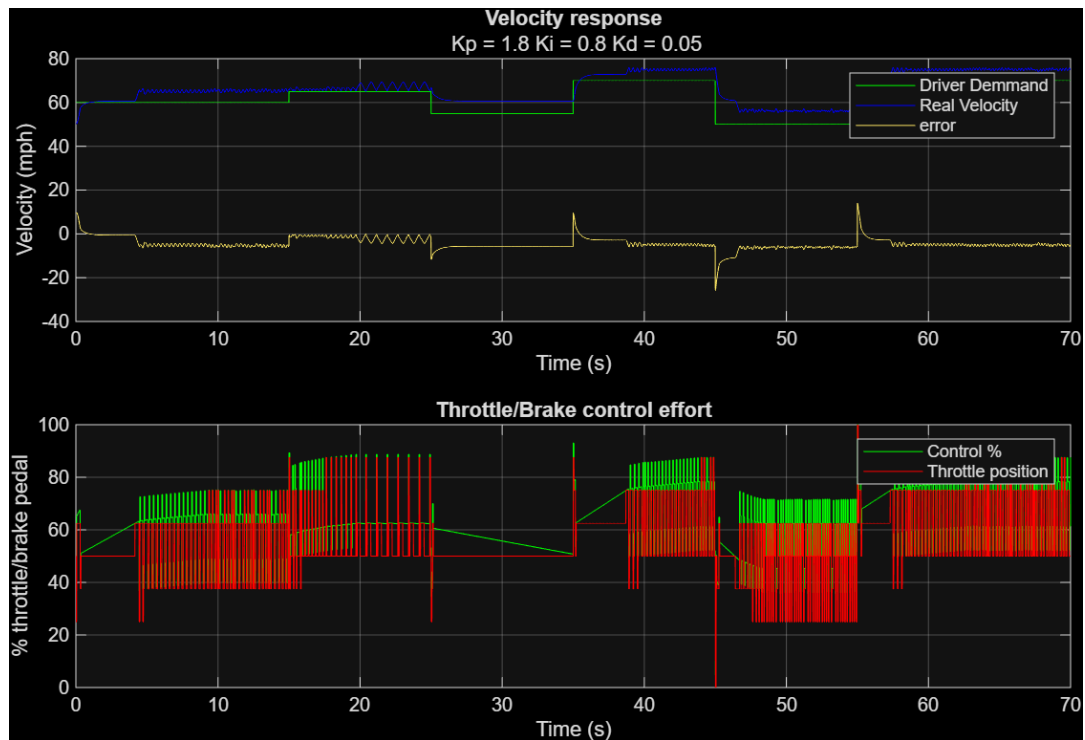


Figure 1. Number of bits equal to 4

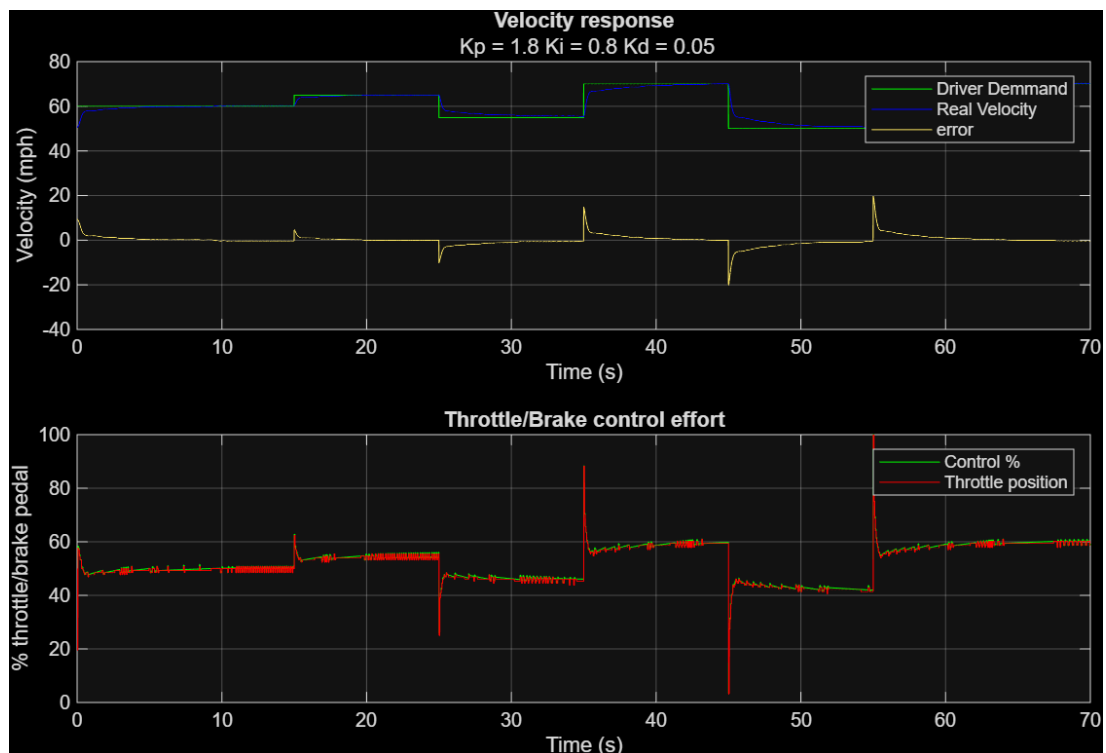


Figure 2. Number of bits equal to 8.

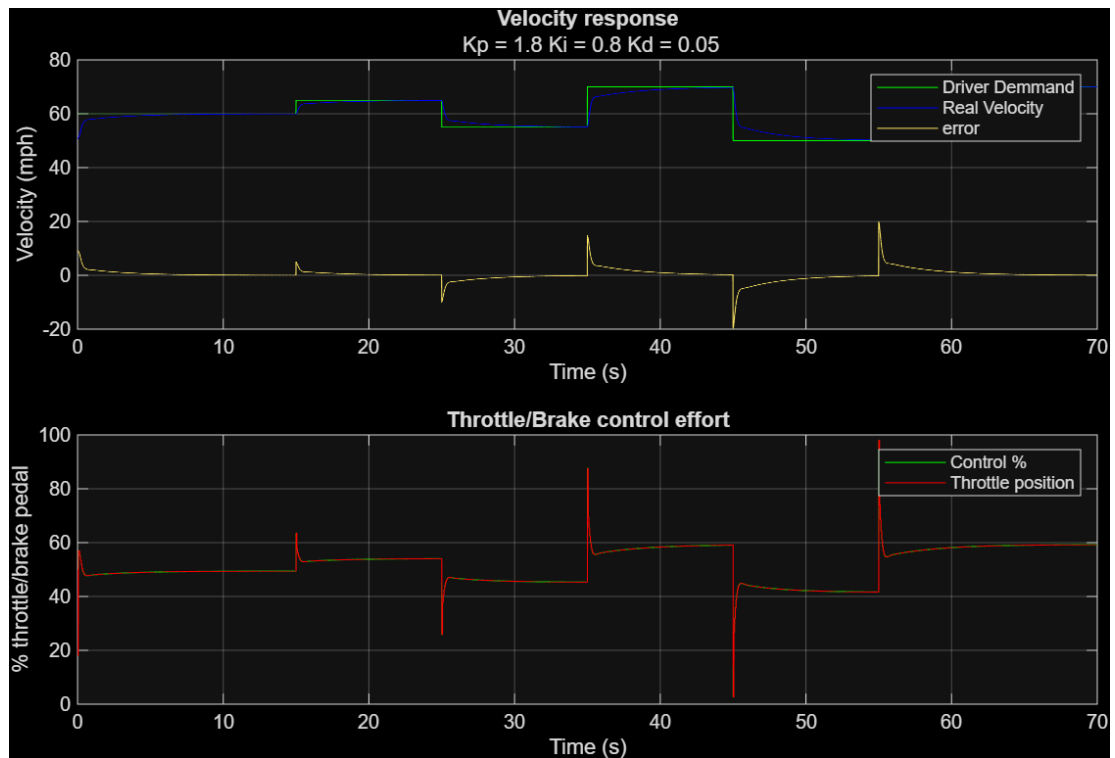


Figure 4. Number of bits equal to 16.

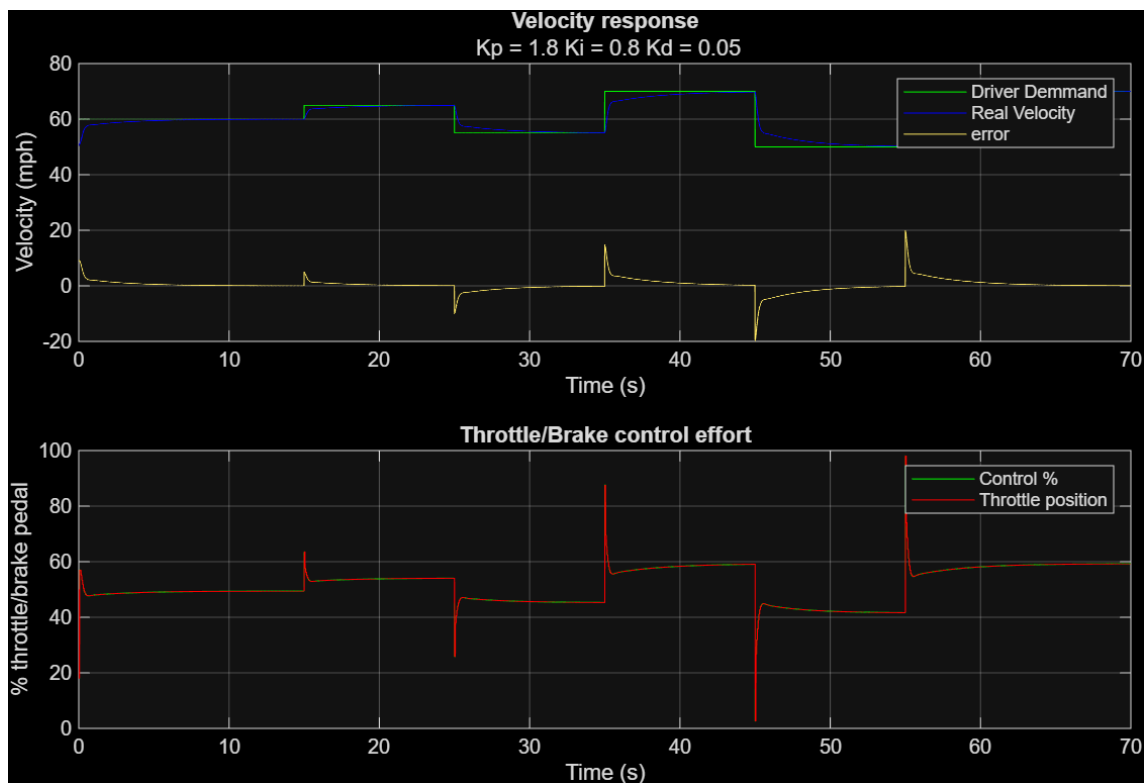


Figure 3. Number of bits equal to 32.

With a lower value of bits for the encoding/decoding process, it will give a quicker response on the control, however the accuracy of the velocity will be much poor, and some significant error is going to maintain during the performance. On the other hand, as the number of bits is increasing, the accuracy increases and achieve the velocity demanded by the driver is possible with a near 0 steady error, although the velocity of the control's response will be slower.

4. Sensitivity analysis of different sampling times.

After analysing a range of sampling times, the longest sample time that gives a reasonable performance is 0.3s, above of that value, it starts to get really noisy and unable to get driver's demand consistently, due to the fact that the signal control tries to get the value in less time so the correction value gets higher and never achieve the demand.

It is important to analyse the sampling time as will be the difference on how frequently the signal control will change. In a real scenario, a high sampling time could lead to a slow response to unexpected changes.

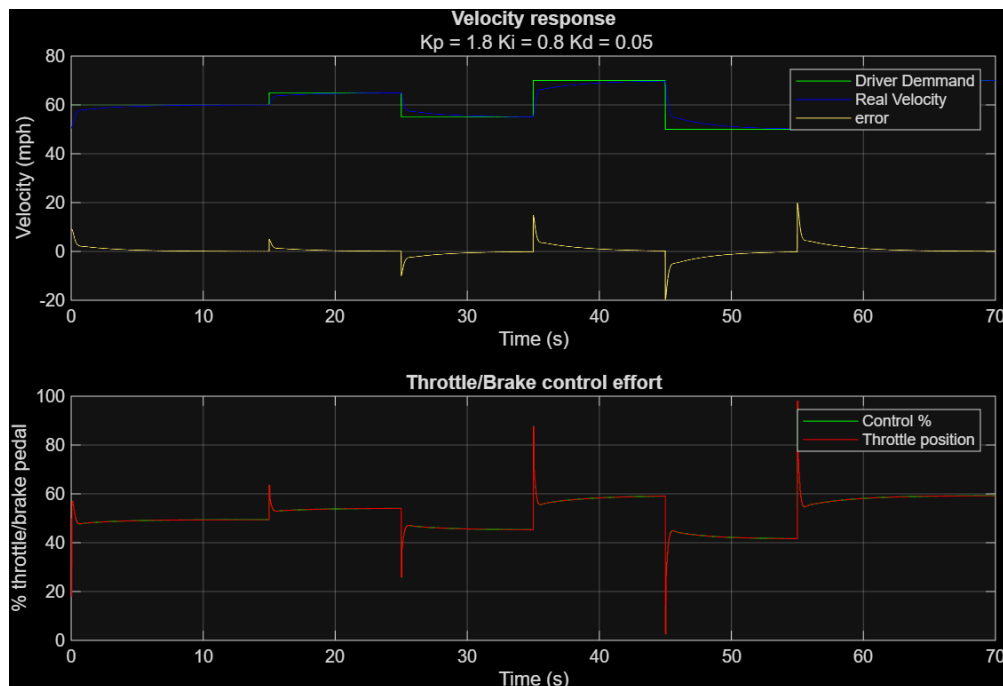


Figure 5. Sampling time equals to 0.05

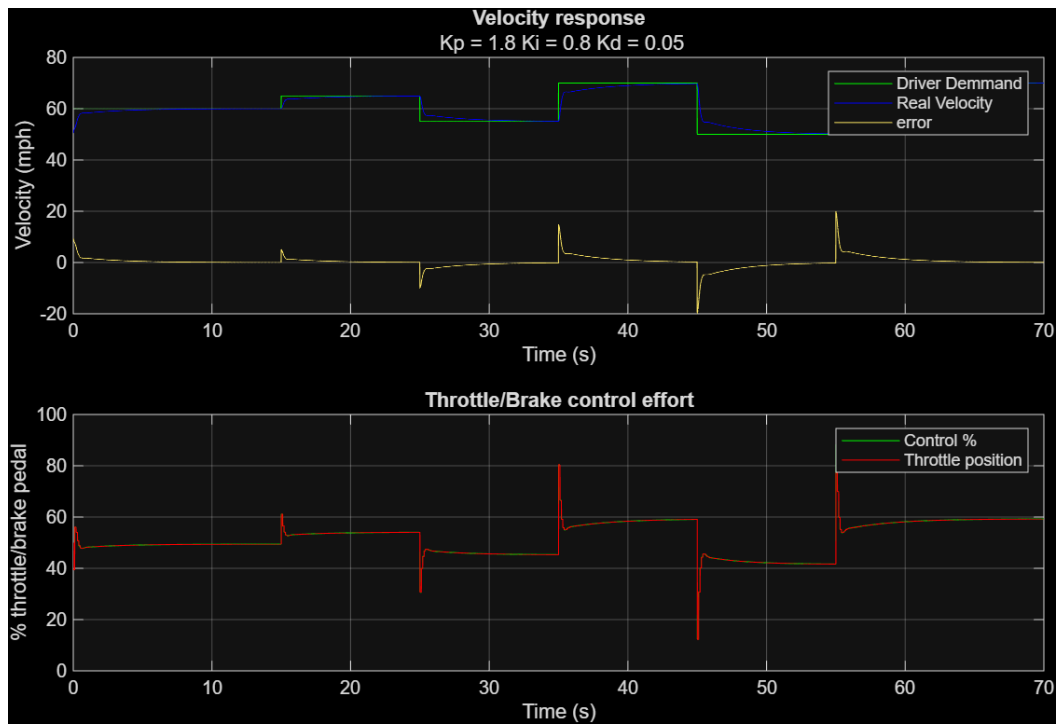


Figure 6. Sampling time equals to 0.1

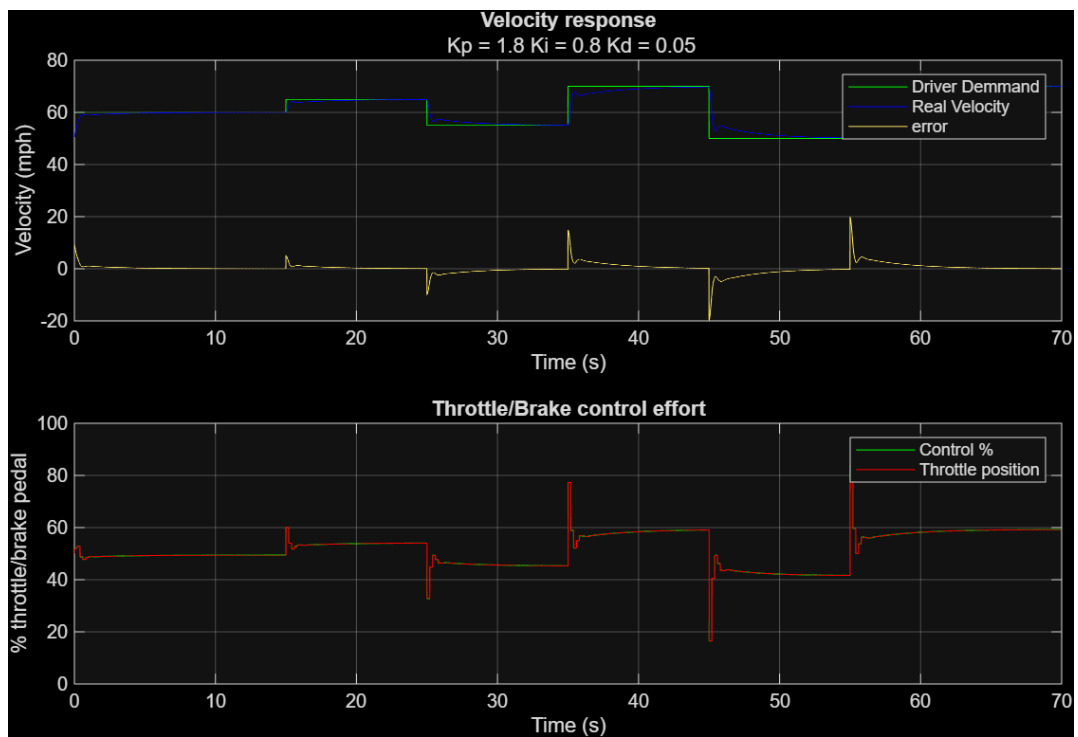


Figure 7. Sampling time equals to 0.2

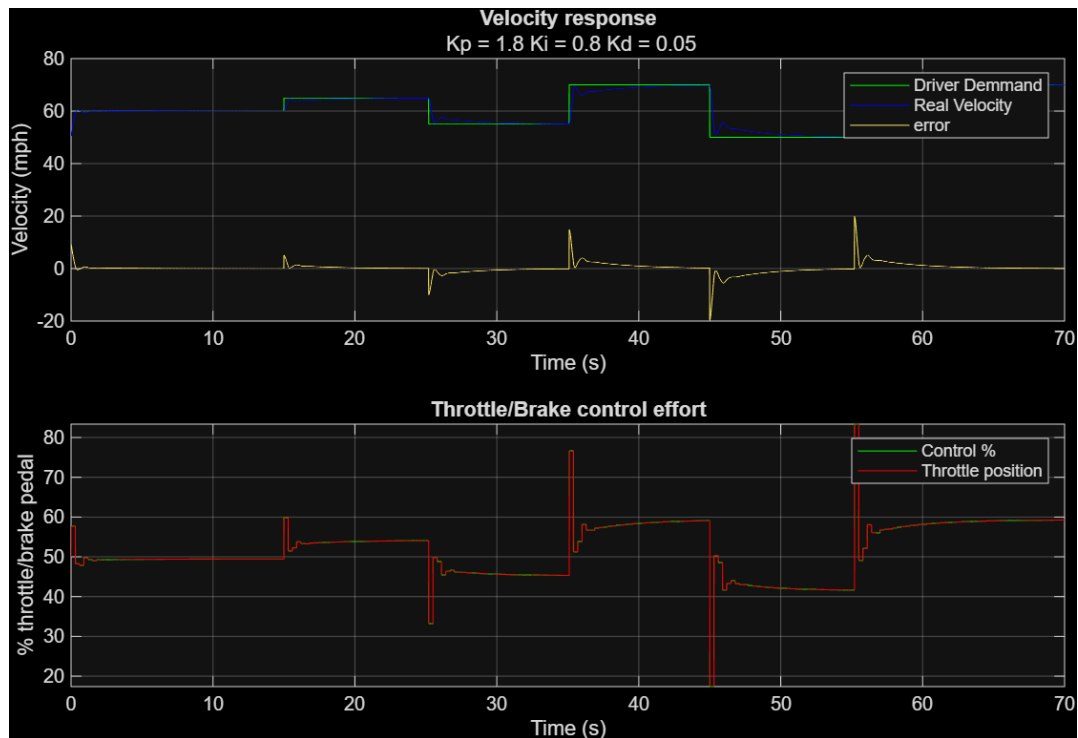


Figure 8. Sampling time equals to 0.3

5. Sensitivity analysis of communication network delay.

Different values for the delay were assigned to commit a sensible study on how this could affect the performance of the control. The maximum value, with the current configuration on the controller gains, that allows an acceptable performance would be 0.05, above from this value, the control signal will go back and forward dropping its accuracy and not manage to achieve driver's demand.

The delay provoke that the signals will take a longer to stabilize the signal and achieve the 0 steady error. If the delay is higher could lead the signal into a loop where it will never achieve the demand and be changing with bigger steps.

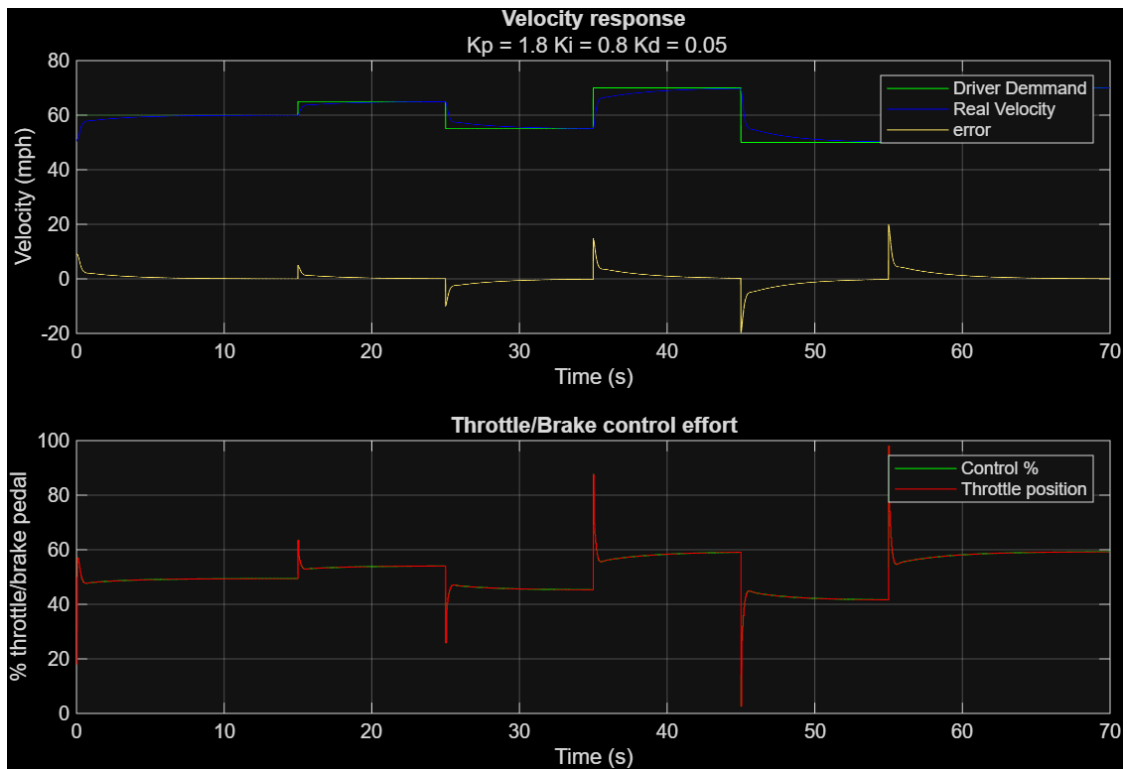


Figure 9. Communication delay equals to 0.00

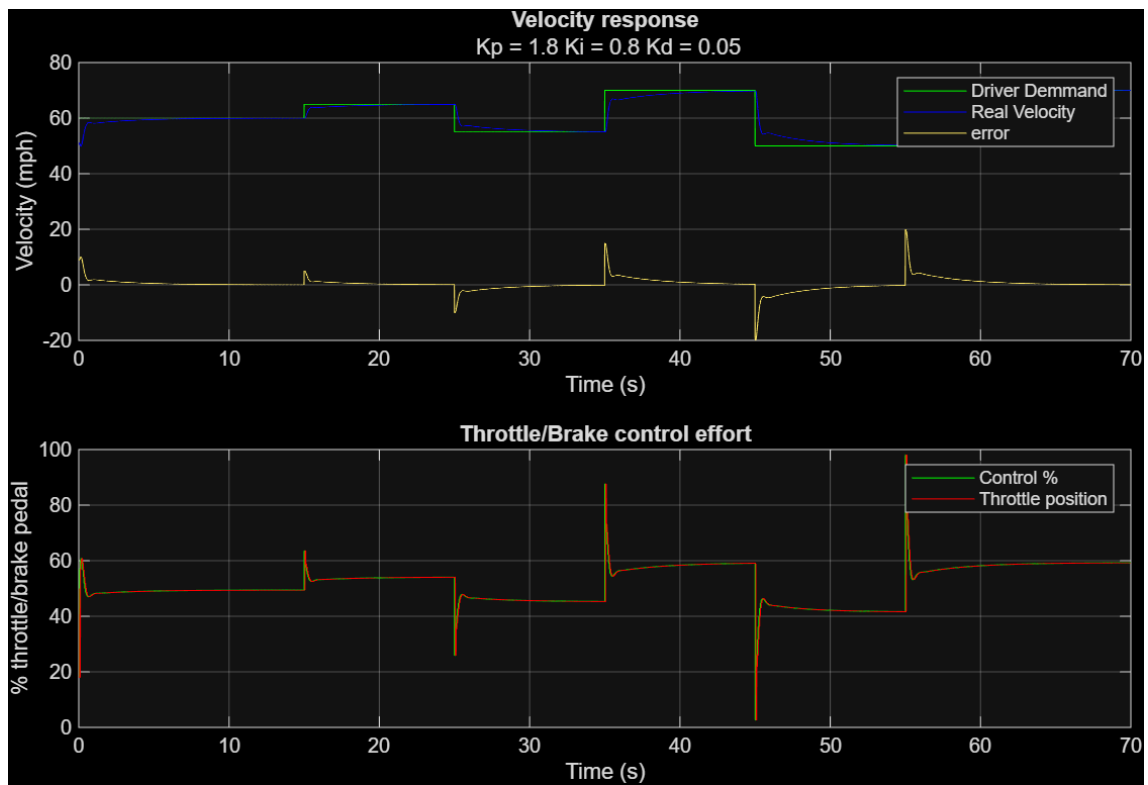


Figure 10. Communication delay equals to 0.05

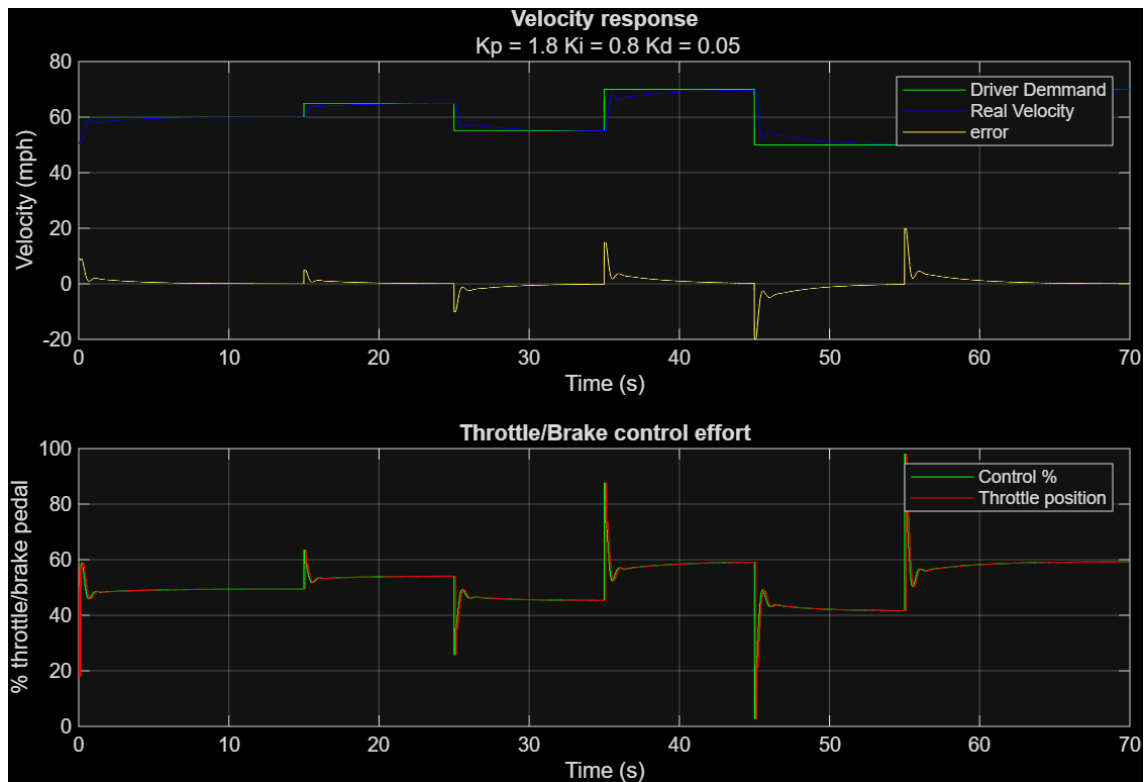


Figure 11. Communication delay equals to 0.06

For the current configuration, it seems that the values of the delay do not have a big effect on the accuracy, which means this configuration should be stable. However, as it was mentioned before, increasing the delay could lead to a loop in the control system and create the need to recalibrate the PID parameters.

6. Conclusions

Comparing with different technical papers that analyse the development of a cruise control system, there is a lot of room for improvement in this code.

For example, the implementation of autonomous iteration for the PID coefficients so they could adapt on real time. This upgrade would help the PID increase its accuracy despite of any change on the parameters or unexpected situations.

Another important aspect to highlight, this cruise control is only considering the driver's demand, however, in the research developed (Zhu, M., and Tan, G., 2023) and the regulations from ISO 15622:2018 (ISO (2018)), they remark the importance of other aspects, for instance, the distance to other vehicles on the road.

In conclusion, despite the actual cruise control code works with a stable steady state error and quick response, it would need important implementations for its viability on a real-life environment to be able to attend the regulations, as the ISO 15622:2018 (ISO (2018)) establish for the adaptive cruise control systems.

7. References.

Mathworks.com. (2019). Convert Text Representing Binary Value. [online] Available at: <https://uk.mathworks.com/help/matlab/ref/bin2dec.html> [Accessed 26 Mar. 2026].

Mathworks.com. (2026). *persistent - Define persistent variable - MATLAB*. [online] Available at: <https://uk.mathworks.com/help/matlab/ref/persistent.html> [Accessed 26 Mar. 2026].

Zhu, M., and Tan, G., "Research on Cooperative Adaptive Cruise Control (CACC) Based on Fuzzy PID Algorithm," SAE Technical Paper 2023-01-0682, 2023, <https://doi-org.uwtsd.idm.oclc.org/10.4271/2023-01-0682>.

Jiang, S., Smith, M., and Kitchen, J., "Optimization of PID Control for Engine Electronic Throttle System Using Iterative Feedback Tuning," SAE Technical Paper 2009-01-0370, 2009, <https://doi-org.uwtsd.idm.oclc.org/10.4271/2009-01-0370>.

Harada, Y., Miyata, H., Hayakawa, Y., and Fujii, S., "Cruise Control System Using Adaptive Control Theory," SAE Technical Paper 931917, 1993, <https://doi-org.uwtsd.idm.oclc.org/10.4271/931917>.

Wuh, D., "Design and Comparison of Autonomous Intelligent Cruise Control Systems," SAE Technical Paper 942405, 1994, <https://doi-org.uwtsd.idm.oclc.org/10.4271/942405>.

Chen, X., Wang, R., Cui, Y., Jin, X., et al., "TD3 Tuned PID Controller for Autonomous Vehicle Platooning," SAE Technical Paper 2023-01-7108, 2023, <https://doi-org.uwtsd.idm.oclc.org/10.4271/2023-01-7108>.

ISO (2018). ISO 15622:2018(E): Intelligent transport systems — Adaptive cruise control systems — Performance requirements and test procedures. [online]

Available at:

<https://cdn.standards.iteh.ai/samples/71515/46b6e07bde1f4c6f91cd143563bdc70f/ISO-15622-2018.pdf> [Accessed 26 Mar. 2026].