

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:
Digital Cruise Controller Implementation and Analysis***

Student name: Aangan Nitinkumar Patel

Student ID number: 2511974

Program: MSc – Automotive Engineering

Engineering Module: Energy Storage and Control Systems

Module code: ACHA7007

Date: 29/03/26

Contents

1. Introduction	2
2. Encoding and decoding of signals	3
3. Feedback controller design and baseline response	4
4. Effect of bit depth on closed-loop performance	5
5. Effect of sampling time	6
6. Effect of communication network delay	7
7. Additional implementation comments	9
8. Conclusion	10
Performance criteria used in the sweep analysis	11
References	12

1. Introduction

Cruise control is often thought of as an easy problem to solve in terms of feedback theory. However, its digital implementation is influenced by factors that go beyond controller gain. When dealing with an automotive ECU, for instance, the controller is only presented with discretely sampled and quantized signals. The controller's actions are then sent to the actuators after some finite delay. This means that resolution and communication delay become an integral part of the controller design and are no longer peripheral software considerations.

In this report, an evaluation of the MATLAB cruise controller design is presented as an all-inclusive digital automotive controller system. The design is based on a PI controller because its integral term has zero steady-state error and is structurally simple to allow for easy identification of effects of quantization, sampling rate, and delay. The analysis is also interpreted with reference to ISO 15622 for longitudinal control performance, ISO 11898 for CAN-based vehicle communication, and ISO 26262 for malfunctioning E/E systems in road vehicles. In this context, controller performance is shaped not only by gain selection, but also by sampling, quantisation and delay, which are well-recognised issues in digital and networked feedback systems (Åström and Murray, 2020; Le Ny, 2011; Liu, Selivanov and Fridman, 2019).

2. Encoding and decoding of signals

The first requirement of the practical problem was to map the received raw binary signals to physical values and then to convert the controller output back to its binary form for transmission to the actuator. For an N-bit unsigned signal, its integer value is within the interval $[0, 2^N - 1]$. The maximum value that can be represented is given by the equation $R = 2^N - 1$. For the MATLAB implementation, `bin2dec` is used to map back to its integer value and then linearly map to its physical value. On the other hand, rounding and saturation are performed, followed by conversion back to its binary form using `dec2bin` (MathWorks, n.d.-c; MathWorks, n.d.-d).

The same approach was followed for the velocity sensor, the driver demand signal, and the actuator command. The reason for the significance of this is the fact that improper scaling or inconsistent limits may lead to implementation faults, which may appear similar to controller faults. Thus, quantization is not only an issue of data formatting, but the signal received by the controller is changed, and the quantization may affect the closed-loop control. The encoder and decoder were thus considered part of the control implementation (Brockett and Liberzon, 2000; Microchip Technology, 2020).

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

Figure 1. Signal encoding and decoding relationships

3. Feedback controller design and baseline response

A discrete proportional-integral (PI) controller was used to ensure zero steady-state errors for the task. The controller processed the tracking error, integrated the error, and then processed the proportional-integral controller to generate the actuator commands. The final gains were selected as $K_p = 2.0$, $K_i = 0.6$, and the sampling period as $T_s = 0.05$ seconds for the baseline scenario. The gains were chosen to provide a balance between the rate of correction and the level of actuator commands. Increasing the proportional gain increases the rate of correction but also increases the level of actuator commands, while increasing the integral gain improves the rate of eliminating the error but can produce more oscillations due to the presence of a delay or coarse quantization (Åström and Murray, 2020; MathWorks, n.d.-b).

The controller implemented actuator saturation, which was defined as ranging from -100% to +100%, with simple anti-windup clamping. Without this feature, the continuous integration during saturation would lead to a longer settling time and increased overshoot as the system returned from the saturated state to the linear region after the saturation had been removed. The results of the baseline 8-bit controller were as expected, with the vehicle speed tracking the control input without instability, and the tracking error converging to zero after each step change in the reference. The accompanying graph shows the expected trade-off between tracking accuracy and control action, as faster correction requires distinct changes in the commands (MathWorks, n.d.-a; MIT, 2004).

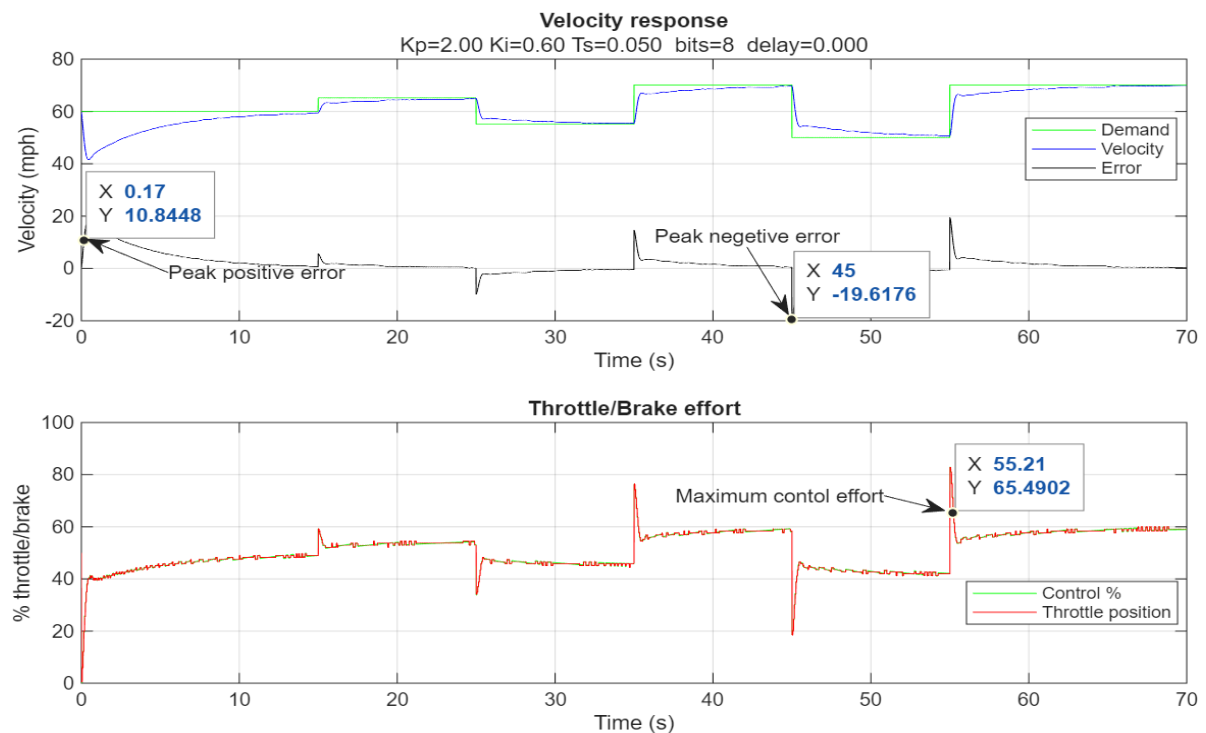


Figure 2. Baseline 8-bit closed-loop response

4. Effect of bit depth on closed-loop performance

The bit depth sweeps also showed that the resolution of the signal had a significant impact on the closed-loop system. From the MATLAB results obtained, it was seen that the system configurations using 3 bits, 4 bits, and 5 bits did not satisfy the set criteria. However, system configurations using 6 bits and beyond satisfied the set criteria. This implies that the minimum resolution of the system should be 6 bits. This is important since it implies that the system does not depend on unrealistic precision (Brockett and Liberzon, 2000; MathWorks, n.d.-c).

The difference between the 5-bit and 6-bit quantization will further clarify the mechanism. As the quantization level is reduced to 5-bit, the measured and commanded signals are restricted to only a limited number of discrete values, and the controller attempts to control the plant based on an approximation of the actual dynamics. This leads to an increased step in the velocity profile and the control efforts. As bit depth increases, the quantization step size decreases, allowing finer signal representation and smoother closed-loop behavior, making it possible for the controller to control the plant within the specified range. Although the minimum error is for the 16-bit case, the difference between the 16-bit and the 8-bit and 6-bit cases is very small, and the 8-bit case is taken as the standard for reporting (Brockett and Liberzon, 2000).

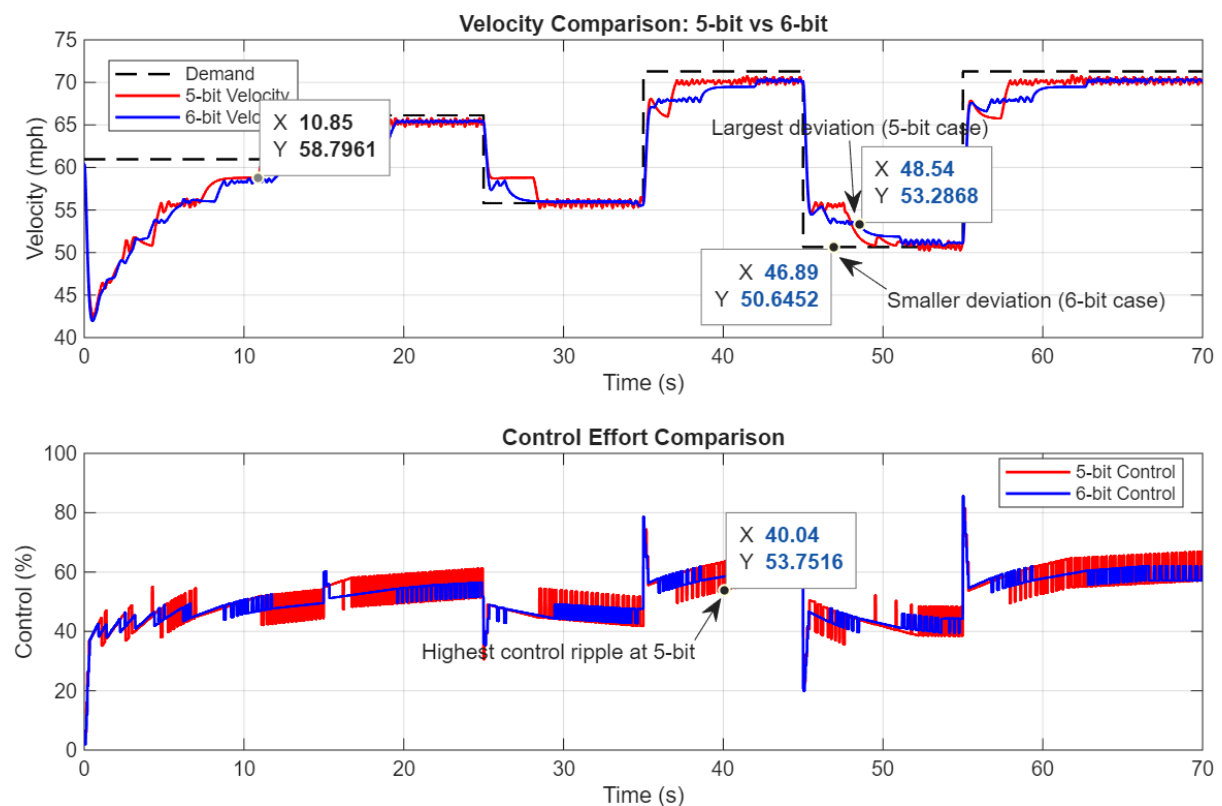


Figure 3. Bit-depth comparison (5-bit vs 6-bit)

5. Effect of sampling time

The sampling-time sweep showed that all tested values between 0.01 s and 0.30 s satisfied the selected performance criteria. This means that 0.30 s was the largest acceptable sampling time within the investigated range, but it does not imply that all larger values would also be acceptable. This interpretation is consistent with sampled-data control theory, since increasing the sampling interval reduces update frequency and increases the adverse effects of sample-and-hold behaviour (Le Ny, 2011).

Increasing the sampling time reduces the rate at which new information is introduced to the controller and increases the effective lag associated with sample-and-hold behaviour (Le Ny, 2011). In an automotive ECU, additional time must also be available for controller execution, sensor updates and network scheduling. Since many such devices communicate over CAN, message timing is also influenced by arbitration and network behaviour (ISO, 2015; Corrigan, 2017). As such, 0.05 s remains the stronger engineering choice in practice.

6. Effect of communication network delay

The communication-delay study produced the clearest pass/fail boundary in the report. Delays from 0 s to 0.25 s all satisfied the selected criteria, but 0.30 s failed because the peak absolute tracking error increased to 23.203 mph, above the 20-mph threshold. At 0.35 s and 0.40 s, the response deteriorated much more severely, with strong oscillation and very high control effort. For the chosen controller tuning and tested conditions, the maximum acceptable actuator delay was therefore 0.25 s. This is consistent with the wider literature showing that delay can reduce damping and degrade closed-loop performance in networked control systems (Liu, Selivanov and Fridman, 2019).

The results for the 0.25 s to 0.30 s delay range clearly show the criteria for passing or failing the test, as the controller still manages to track the demand at 0.25 s, while at 0.30 s, the oscillation is high, indicating that the system is not damp enough. The control effort graph confirms the results, as the delayed loop still responds to the changes, but the fact that the information is delayed by 0.30 s results in the system repeatedly "hunting" to find the correct control action. The results are also relevant from a standardization viewpoint, as the ISO 11898 standard defines the CAN protocol as a real-time distributed communication architecture for road vehicles, indicating that the results of a delayed system are more than a simple mathematical inconvenience, but a problem in the actual system design. The ISO 15622 standard emphasizes the minimum functionality and performance testing of longitudinal control systems, indicating that the lack of damping at 0.30 s would be a problem in a real-world test, as opposed to a simulation. While other gain values might allow for a slightly longer delay, this would actually reduce the bandwidth of the loop, which would then adversely affect the responsiveness of the loop (Liu, Selivanov and Fridman, 2019; Åström and Murray, 2020).

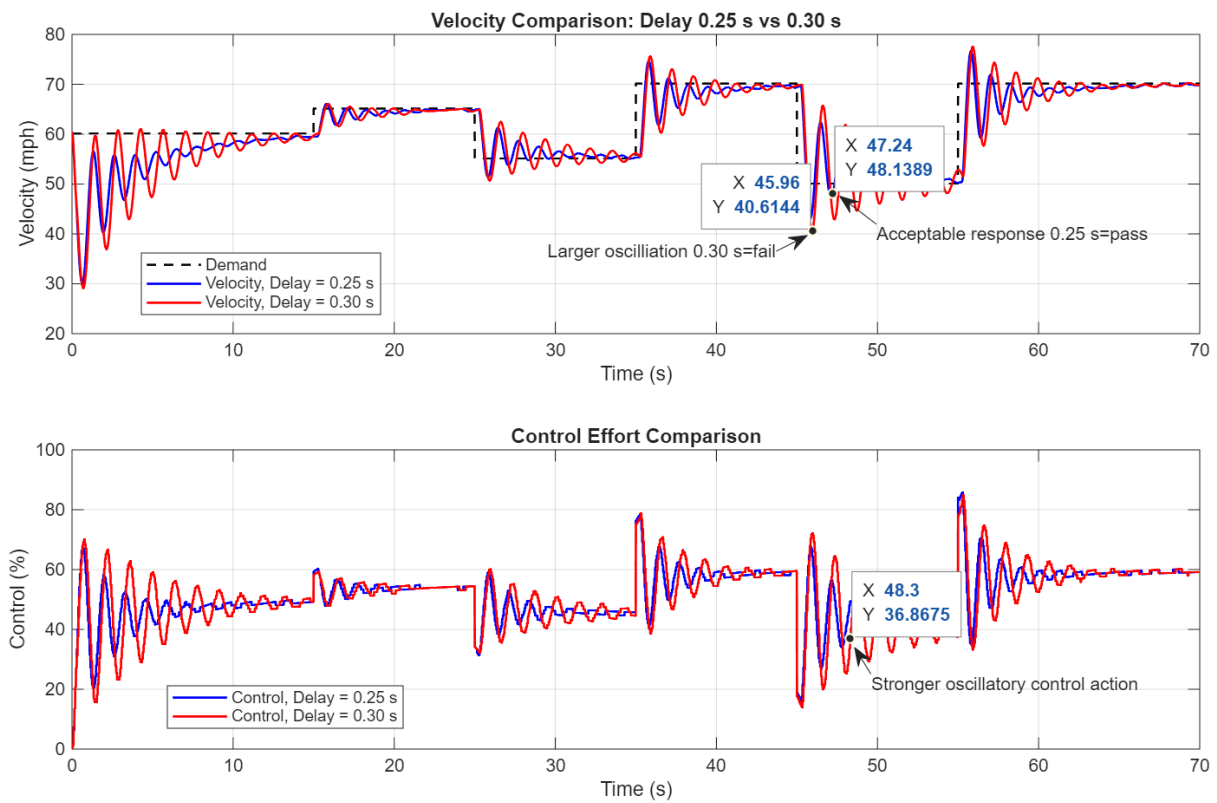


Figure 4. Actuator-delay boundary response (0.25 s vs 0.30 s)

7. Additional implementation comments

A significant advantage of the MATLAB approach was the consideration of controller design and implementation as a single problem. The system's performance was not solely dependent on the correctness of the proportional-integral (PI) relationship but also on signal scaling, realistic limitation of the actuator command, consideration of wind-up, and evaluation of system response over a range of bit depths, timing, and delay.

Although the MATLAB approach can be seen as a limited validation process, the results serve to illustrate the benefits of a standards-based approach. Issues such as road gradient, sensor noise, bus jitter, and ECU scheduling were not included within the approach. Therefore, the results provided are specific to this plant model and gain configuration. However, the approach taken highlights the importance of quantization, sampling time, and delay on system predictability, which can be seen as relevant to the overall ISO 26262 process.

8. Conclusion

Overall, the results showed that digital implementation choices are as important as controller structure in determining closed-loop performance. For the selected plant model and PI tuning, 6 bits was the minimum acceptable signal resolution, 0.30 s was the largest acceptable sampling time within the tested range, and 0.25 s was the maximum acceptable actuator delay before the response became unacceptable. These findings align with the broader literature on quantisation, sampled-data control and delay-sensitive feedback systems (Åström and Murray, 2020; Brockett and Liberzon, 2000; Le Ny, 2011; Liu, Selivanov and Fridman, 2019).

Performance criteria used in the sweep analysis

The same performance criteria were applied throughout the bit-depth, sampling-time and delay sweeps so that pass/fail decisions were consistent, transparent and repeatable.

Table 1. Performance criteria used for sweep-analysis pass/fail assessment

Criterion	Threshold
Mean absolute error in final 3 s	≤ 1.0 mph
Peak absolute error after first 1 s	≤ 20.0 mph
Peak control effort	$\leq 95\%$
Stability	No blow-up or non-finite values

References

- Åström, K.J. and Murray, R.M. (2020) *Feedback systems: An introduction for scientists and engineers*. 2nd edn. Available at: https://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-public_24Jul2020.pdf (Accessed: 25 March 2026).
- Brockett, R.W. and Liberzon, D. (2000) 'Quantized feedback stabilization of linear systems', *IEEE Transactions on Automatic Control*, 45(7), pp. 1279–1289. Available at: <https://hrl.harvard.edu/publications/brockett00quantized.pdf> (Accessed: 25 March 2026).
- Corrigan, S. (2008) *Controller area network physical layer requirements*. Texas Instruments Application Report SLLA270. Available at: <https://www.ti.com/lit/pdf/slla270> (Accessed: 25 March 2026).
- Corrigan, S. (2017) *Introduction to the controller area network (CAN)*. Rev. B. Texas Instruments Application Report SLOA101. Available at: <https://www.ti.com/lit/pdf/sloa101> (Accessed: 25 March 2026).
- ISO (2011) *ISO 26262-1:2011(en), road vehicles — functional safety — Part 1: Vocabulary*. Available at: <https://www.iso.org/standard/43464.html> (Accessed: 25 March 2026).
- ISO (2015) *ISO 11898-1:2015, road vehicles — controller area network (CAN) — Part 1: Data link layer and physical signalling*. Available at: <https://www.iso.org/standard/63648.html> (Accessed: 25 March 2026).
- ISO (2018) *ISO 15622:2018, intelligent transport systems — adaptive cruise control systems — performance requirements and test procedures*. Available at: <https://www.iso.org/obp/ui/en/> (Accessed: 25 March 2026).
- Le Ny, J. (2011) *Sampling and sampled-data systems*. Course notes. Available at: https://www.professeurs.polymtl.ca/jerome.le-ny/teaching/NECS_Spring11/notes/3_sampling/3_sampling.pdf (Accessed: 25 March 2026).
- Liu, K., Selivanov, A. and Fridman, E. (2019) 'Survey on time-delay approach to networked control', *Annual Reviews in Control*, 48, pp. 57–79. Available at: https://eprints.whiterose.ac.uk/id/eprint/153531/1/NCS_Review_paper_R1_20190430.pdf (Accessed: 25 March 2026).

MathWorks (n.d.-a) *Anti-windup control using PID controller block*. Available at: <https://www.mathworks.com/help/simulink/slref/anti-windup-control-using-a-pid-controller.html> (Accessed: 25 March 2026).

MathWorks (n.d.-b) *PID controller – continuous-time or discrete-time PID controller*. Available at: <https://www.mathworks.com/help/simulink/slref/pidcontroller.html> (Accessed: 25 March 2026).

MathWorks (n.d.-c) *Quantization*. Available at: <https://www.mathworks.com/help/fixedpoint/quantization.html> (Accessed: 25 March 2026).

MathWorks (n.d.-d) *Quantization (Fixed-Point Designer user guide)*. Available at: <https://www.mathworks.com/help/fixedpoint/ug/quantization.html> (Accessed: 25 March 2026).

MathWorks (n.d.-e) *Prevent block windup in multiloop control*. Available at: <https://www.mathworks.com/help/simulink/slref/prevent-block-windup-in-multiloop-control.html> (Accessed: 25 March 2026).

Microchip Technology (2020) *Enhanced controller area network (CAN) module family reference manual*. Available at: <https://ww1.microchip.com/downloads/en/DeviceDoc/dsPIC33E-PIC24E-FRM-Enhanced-Controller-Area-Network-CAN-DS70000353D.pdf> (Accessed: 25 March 2026).

MIT (2004) *2.737 Mechatronics Laboratory Assignment 1: Servomotor Velocity Control*. Available at: https://web.mit.edu/2.737/www/labs/Lab1/Lab1_S061.pdf (Accessed: 25 March 2026).

Robert Bosch GmbH (1991) *CAN specification 2.0*. Available at: <https://esd.cs.ucr.edu/webres/can20.pdf> (Accessed: 25 March 2026).

Texas Instruments (2013) 'Basics of debugging the controller area network (CAN) physical layer', *Analog Applications Journal*. Available at: <https://www.ti.com/lit/pdf/slyt529> (Accessed: 25 March 2026).