

Control System Modelling

Joseph McBride

ACMA6001 Control Technology

29/09/23

Table of Contents

Table of Figures	2
Task 1.....	3
Introduction	3
System Control Functions	4
Sampling Rates & Performance	5
Communication Protocols.....	5
Software Development & Analysis Tools.....	6
Control System Improvements.....	6
Task 2.....	7
Description of Encoding and Decoding Functions	7
Feedback Controller Response.....	7
Bit Range Experimentation	9
Sample Rate Experimentation.....	9
Additional Considerations	10
Bibliography	11

Table of Figures

Figure 1 - Index to Encoder Body	3
Figure 2 - PCB to Encoder Body.....	3
Figure 3 - Decoding Function	7
Figure 4 - Encoding Function.....	7
Figure 5 - Initial Cruise Control Response	8
Figure 6 - Revised Cruise Control Response	8

Task 1

Introduction

Within industry control system are used to perform a variety of applications. In this specific example the control system being analysed is utilised during the manufacturing process of an encoder. More specifically, the system is responsible for bonding the index and PCB to the encoder body through means of an automation process. Figure 1 and Figure 2 illustrate the assembly process of the encoder within the control system.

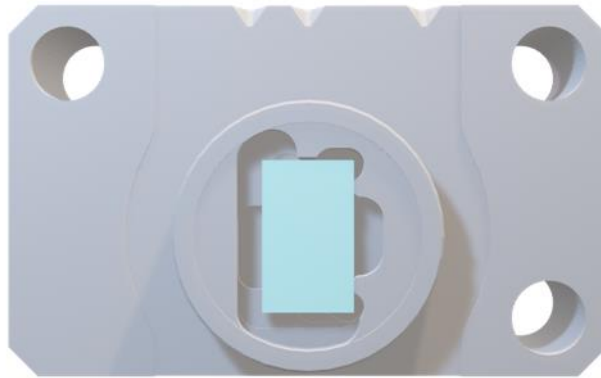


Figure 1 - Index to Encoder Body

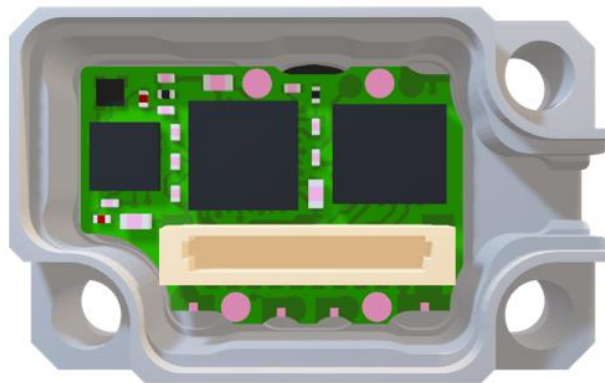


Figure 2 - PCB to Encoder Body

The system is PLC controlled and uses several sensors and actuators to achieve the desired output. Examples of some of the sensors incorporated into the system include infra-red type for part detection purposes and reflective laser for distance detection applications. A selection of actuators are integrated including a 6-axis robotic arm, pneumatic pistons, and servo drives. The sensors previously mentioned are the inputs to the system which enable monitoring of the machine status whereas the actuators are the outputs which allow the machine to perform its intended functions.

In the following sections of this report, the control system will be analysed in more detail. This analysis will take a closer look at the control functions utilised within the system. Due to the vast nature of the automation system, there are various functions that are often carried out simultaneously. The system performance and software development tools will also be discussed in the following section.

System Control Functions

As previously mentioned, the control systems purpose is to bond the index and PCB to the encoder body. This takes place within an automation cell with several assembly stations that carry out an assembly operation. The product is transferred around the cell using a 6-axis robotic arm. Parts transferred into the cell via a conveyor system. Trays are used to secure the encoder bodies into place when travelling on the conveyor. Encoder bodies will be unloaded from the trays when they enter the cell and then loaded back into the trays once the assembly process has been completed. Figure 3 below shows the model of the control system in question.

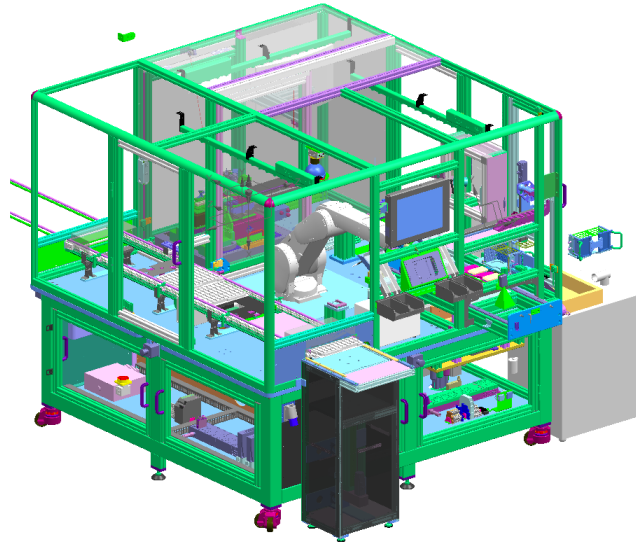


Figure 3 - Encoder Assembly Automation Cell

One of the assembly stations within the automation cell is the 'PCB Router'. This is where the individual PCBs are cut from the panel before being placed into the encoder body by the robotic arm. Vision systems are used within the process to determine the PCB tab positions which will need to be cut with the router spindle. In this case the vision system is mounted on the robotic arm so that it can be positioned over the PCB panel. Once the tab positions have been determined they are communicated back to the PLC using the SLMP communication protocol. This is an ethernet based protocol which is often used to communicate with PLCs. After the data has been received by the PLC it can then perform a series of calculations to determine the tab coordinates. Shown below in are an example of the tab position calculations within the PLC ladder code. These coordinates are then sent to the servo motors to move the spindle into position and cut the tab.

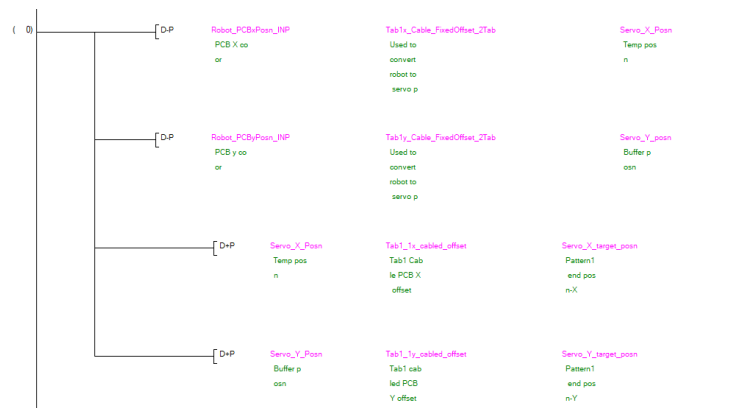


Figure 4 - PCB Tab Position Function

Sampling Rates & Performance

Sampling rates are an important factor when it comes to control system performance. If the rate in which the system samples inputs is incorrect then it is likely that an input to the system could be missed. One of the parts of the encoder assembly control system that is heavily reliant on sampling rates is the conveyor. There are four proximity sensors positioned along the conveyor, these sensors are positioned as follows:

- *Tray Entrance to Machine*
- *Tray Entrance to Working Area*
- *Tray Leaving Working Area*
- *Tray Leaving Machine*

Due to the velocity of the tray moving along the conveyor the sampling rate needs to be set appropriately to capture the tray status at each of these points. “A common rule of thumb is to sample signals at 10 times the bandwidth of the system or the bandwidth of interest for the model” (National Instruments, 2023). As this is a PLC controlled system it is crucial that the input module selected meets the specification of the sampling rate required. For this application the Mitsubishi ‘QX80’ input module is utilised. The control system performance would be poor if the sampling rate resulted in inputs not being captured. The sampling rate will be set at the CPU model using the necessary software.

Another part of the system reliant on sampling rates is the picking and placing of parts by the robotic arm. A signal from the robot controller will be sent to the PLC to confirm the status of the gripper. The speed of this application needs to be taken into consideration so that the robot does not perform the move operation before the part is gripped correctly.

Communication Protocols

The hardware implanted throughout the control system requires a few different communication protocols to be used throughout. One of the main communication protocols used is CC-Link, the reasoning being that much of the hardware is manufactured by Mitsubishi. CC-Link (Control and Communications Link) “is a fieldbus system developed by Mitsubishi in 1996” (Jäger, 2020). The transmission speed, of the CC-Link network, in this control is set to 10Mbit/s. One of the advantages of CC-Link is that it allows for communication with several sensors and actuators with minimal wiring required.

Most sensors and actuators in the system are physically wired to a Festo CPX module. The module is made up of input and output blocks, it also contains banks of pneumatic actuators. CC-Link is then used to link this module to the PLC so that the inputs can be monitored, and the outputs can be controlled. Another communication protocol utilised on the encoder assembly system is Ethernet/IP. This is predominately used within the control system for communication with the various vision systems implemented throughout the automation cell. Static IP addresses are allocated to all the devices so that they can be accessed main controller.

Software Development & Analysis Tools

As mentioned previously, the hardware implemented within the control system is produced by various manufacturers. Therefore, different software development tools are required to program the devices. The PLC program is developed using 'GX Works 2'. This is a PLC programming software development tool from Mitsubishi "that enables projects to be created at the system overview stage has been added" (Mitsubishi Electric, 2023). For this application, the PLC is programmed using SFC (Sequential Function Charts) which each contain ladder logic code. 'GT Designer 3' is the software tool used for HMI programming. This is also a program developed by Mitsubishi which allows for easy integration of their hardware. For robotic programming, 'RT Toolbox' is used. 'MELFA-BASIC' is the programming language used within the development tool. This is also a Mitsubishi application.

Other software development tools used for this control system include Pilz 'PNOZ Multi Configurator'. This is used to program the functional safety aspect of the system. For vision system software development, the Cognex 'In-Sight Explorer' software is used. Programs can be created for specific vision inspection purposes. In terms of analysis tools, the metrics of the control system are recorded into an SQL database. This allows for them to be analysed and manipulated using a variety of queries. It also means that results are backed up for traceability purposes.

Control System Improvements

Identifying continuous improvements is important when it comes to enhancing the performance of a control system. One of the recent improvements implemented onto the system is the addition of a combination of servo motors and actuators to automate the loading and unloading of parts from the cell. Overall, this improvement increases the efficiency of the control system while maintaining minimal operator intervention. More specifically, the improvement automates the loading of PCBs into the router and the unloading of parts trays.

Another improvement that could be made to the control system is to add an additional sensor that would be able to detect the adhesive levels so that it could be monitored. Currently the machine will only recognise when the adhesive has been used completely by the number of dispensing operations that have been performed. One of the challenges presented with this improvement is the adhesive is stored in a tube that is not transparent. Therefore, sensing the level of adhesive inside the tube is difficult. A capacitive sensor could be potentially used to carry out this task, however, it may not obtain the most accurate results.

Task 2

The following section of the report breaks down multiple aspects of the 'Cruise Control' model created in MATLAB. Using the functions available the MATLAB model simulates the response of the controller system with the capability of adjusting the input values to determine the ideal response.

Description of Encoding and Decoding Functions

Section 4 of the 'Cruise Control' model is titled the 'Simulation Loop'. In this part of the code there are encoding and decoding functions. Firstly, the decode function is used to convert the binary data, that would be received from the vehicles ECU, into a decimal format so that it can be processed accordingly. The raw demand element in this function gets the drivers raw velocity, this would be carried out using sensors on the car to determine how quickly the vehicle is moving. The value is then scaled based on the maximum and minimum demand parameters of the system before ensuring that it falls within the appropriate range.

```
% decode raw signal (from x-bit binary to mph)
demand = ((bin2dec(demand_raw).*(M_demand-m_demand))/R)+m_demand;
```

Figure 5 - Decoding Function

Once the drivers demand has been decoded into a decimal format it can then be used in the controller calculations. The controller is responsible for calculating the response based on several different input parameters to the system. After the system response has been calculated it is then encoded so that it can be sent back to the ECU of the vehicle. This is done by scaling the control value to be within a specific range using linear transformation and ensuring it falls within the maximum and minimum allowable control signal values. A decimal to binary translation is then performed so that a control command can be sent to the cars control system. The actuator response, which would be the throttle and brake in this case, can then be adjusted accordingly to obtain the desired output of the system.

```
% encode control signal into x-bit binary number
control_raw = dec2bin(((control-m_ControlSig)/(M_ControlSig-m_ControlSig)).*R);
```

Figure 6 - Encoding Function

Feedback Controller Response

In between the encoding and decoding functions is the feedback controller response. This is a function created in MATLAB to process the raw data input and determine an appropriate response. The main elements input into this function are the velocity, control, and demand values. Also input into this function are the P, I and D values for the PID control aspect of the system. Error is calculated within this function as the difference between the demand and current velocity. There is an 'if' statement in the function that specifies to use the 'control' value if 'u_vec' is less than 'dist_int_diff' and similarly when 'u_vec' is greater than 'dist_int_diff' the code will find the 6th from last value to use in the controller response calculations.

For the proportional, integral, and derivative parts of the function the values are calculated using the parameters input to determine the gain of each factor respectively. The control output is then calculated by adding each of the PID elements together. Finally, the controller function will check if the output value exceeds the upper and lower limits that have been defined for the control signal value. The controller response and error will then be fed out from the function into the main program.

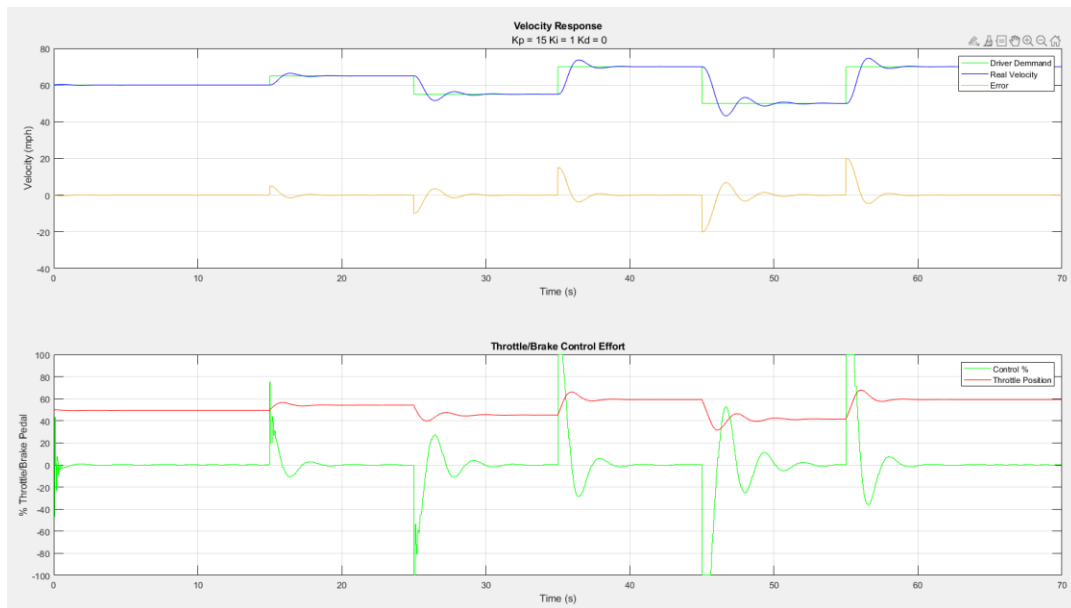


Figure 7 - Initial Cruise Control Response

Shown in Figure 7 is the response of the 'Cruise Control' system using the predefined input parameters. From this response graph it can be determined that there is an aspect of overshoot regarding the 'Real Velocity' compared to the 'Drivers Demand'. This would mean that in some cases the driver may be over the speed limit before the response settles to the required velocity. Another conclusion that can be drawn from the response graph is that the 'Control %' sometimes exceeds the maximum capability as it surpasses 100%. In a real-world environment this would not be attainable therefore, the controllers gain needs to be adjusted to achieve reasonable performance of the system.

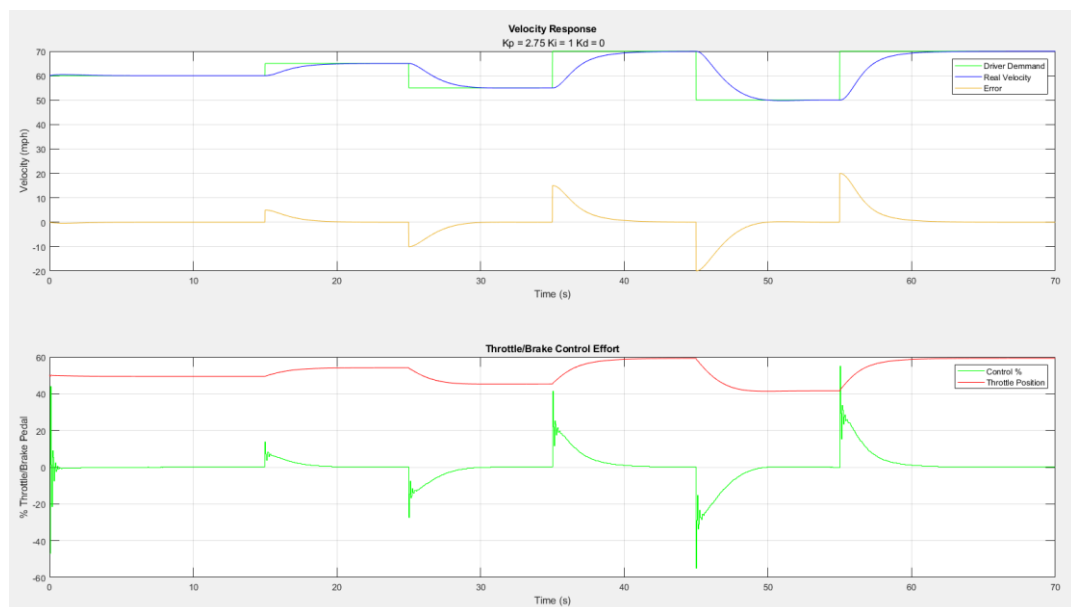


Figure 8 - Revised Cruise Control Response

After adjusting the controller gain (K_p) the above response graph shown in Figure 8 was obtained. The overshoot element in the revised response has been eliminated however, the trade off to this is that the error has increased which means the system takes longer to respond to the 'Drivers Demand'. Even though the error has increased the vehicle would not exceed the 'Drivers Demand' which is more appropriate when compared to real-world application. Also, adjusting the gain value

has limited the 'Control %' from reaching its maximum capability which is another positive. It may be possible to achieve a more reasonable response with further investigation however, in terms of sound engineering judgment, the response achieved in Figure 8 is satisfactory.

To manipulate the response of the system even further the integral and derivative inputs to the controller could be adjusted further. For example, if the integral value was to be increased it may respond to the error seen at a quicker rate however, this could also result in more instability in the velocity response. If the derivative element were to be increased, it would make the system respond more aggressively to the error fluctuation. This would likely increase the 'Control %' as a result due to the throttle and brake needing to be applied more vigorously. Based on these assumptions of further changing input parameters, it is possible to say that the current response achieved a reasonable.

Bit Range Experimentation

It is possible to experiment with the number of bits used for the signal encoding and decoding functions. The MATLAB code uses bits for the actuator, demand, and sensor signals. Currently number of bits used for each aspect is set to 16. In these circumstances it would be appropriate to experiment with an increased and decreased number of bits to see how it affected the response of the 'Cruise Control' system. When increasing or decreasing the number of bits for all three elements there is no change seen to the response of the system. For example, increasing from 16 to 32 bits did not affect the response and decreasing to 8 bits did also not affect the response. When decreasing one of the elements of the cruise control system to 8 bits and leaving the remaining at 16 bits this caused an uncharacteristic response. This means the system did not function as intended.

Reflecting on the point above, by decreasing only one of the elements the system is unable to respond as the responses are not relative to each other. If the bit range of the actuator were to be decreased, then it is unable to respond accordingly to the sensor and demand aspects. This is also the case for the sensor and demand elements when the bit range does not align. Selecting the appropriate bit range for the system depends on several different factors. One of the factors would be the control output range to the vehicles ECU. If these do not align, then there would be a breakdown in communication.

Sample Rate Experimentation

Like the bit range, the sample rate can also be experimented with in the model to achieve an improved response rate of the system. It is likely that an increased sampling rate would improve the response of the system due to the number of cycles that will be performed. For instance, an increased sampling rate would mean the system is able to respond to the changes in demand at a quicker rate. An increased sampling rate depends on the system hardware. If the hardware is capable of coping with the demand of the increased rate. If the sampling rate of the system were to be decreased, then the response rate would also correspond accordingly resulting in poorer system performance.

As mentioned previously, in a real-world scenario the sampling rate would be dictated by the specification of components being used within the system. For a 'Cruise Control' application it is likely that a reasonably sufficient sampling rate could be achieved. This is due to the nature of the system which will need to take into consideration of safety factors. Also, the capability of modern-day vehicle technology would allow for a high degree of precision.

Additional Considerations

There are various factors that have not been considered within the 'Cruise Control' simulation model. One of these factors being the terrain on which the vehicle is travelling on. The system would need to respond accordingly to inclines and declines in the road. If this has not been considered, then an inaccurate response from the simulation model would be received. Many of the newer vehicles on the market use 'Adaptive Cruise Control' which also incorporates additional factors into the control such as the distance to the vehicle or obstruction ahead. The more conventional 'Cruise Control' system just responds to the driver's demand.

To conclude this section, the implementation of the digital controller within the 'Cruise Control' system resulted in the simulation being able to model the overall response. Experimentation with different inputs to the system, such as PID gains and sampling rate, presented a variety of outcomes in relation to the response. With additional investigation and understanding it would be possible to achieve a more accurate simulation model which takes into consideration real-world environment factors.

Bibliography

Jäger, N., 2020. *Igus Blogs*. [Online]

Available at: <https://blog.igus.eu/what-is-cc-link/>

[Accessed 20 September 2023].

Mitsubishi Electric, 2023. *Mitsubishi Electric Factory Automation*. [Online]

Available at: https://gb.mitsubishielectric.com/fa/products/cnt/plc/plceng/gx_works2

[Accessed 20 September 2023].

National Instruments, 2023. *National Instruments*. [Online]

Available at: https://www.ni.com/docs/en-US/bundle/labview-control-design-and-simulation-module/page/lvsysidconcepts/si_sampling_rate.html#:~:text=A%20common%20rule%20of%20thumb,of%20interest%20for%20the%20model.

[Accessed 20 September 2023].