ELECENG 3EY4: Electrical System Integration Project

Lab05_Electronic Speed Controller

Junbo Wang – wangj430 – 400249823

Preet Batra - batrap5 - 400341704

Yichen Lu - luy191 - 400247938

Objective 1: Review of the Lab Manual

For Question 1-4, Yichen Lu led it.
For Question 5-8, Junbo Wang led it.
For Question 9-12, Preet Batra led it.
For Question 13, we equally contributed to it.

Objective 2: VESC Driver Setup

Question 1: What is the role of q-axis current in the operation of a permanent magnet synchronous motor? What is the effect of q-axis current on torque generation?

The primary function of the q-axis current is to generate torque within the motor. When electric current flows through the q-axis of the stator, it interacts with the permanent magnet's magnetic field on the rotor, resulting in the creation of torque. More specifically, this torque is responsible for driving the rotor and performing the mechanical work required by the motor.

The q-axis current in a Permanent Magnet Synchronous Motor (PMSM) directly influences torque generation. By flowing through the q-axis of the stator, it interacts with the rotor's magnetic field, producing torque. The magnitude of the q-axis current determines the strength of this interaction, impacting the motor's ability to generate torque. Optimizing the q-axis current is crucial for achieving efficient and controlled torque output in a PMSM.

Question 2: What is the difference between installing the VESC package directly with apt-get and cloning from GitHub?

Installing the VESC package directly with apt-get provides a stable, system-integrated version which simplifies the process and ensures compatibility. In terms of cloning from GitHub, it offers the latest development version which allows the customization and access to cutting-edge features but requires manual dependency management. As a result, the choice depends on stability preference and the need for the latest developments.

Question 3: Explain what the static_cast in the code is used for. Why is it necessary to cast these values?

The static_cast in the code is used for explicit type casting in C++. It converts the value of myDouble from the double data type to an int. This casting is necessary because if there is not casting, there would be a type mismatch for myInt.

Question 4: Describe what each line of the added code block in Figure 1 does? How does it contribute to parsing the q-axis current data? Why do you think the bytes are shifted (<<) by 24, 16, and 8 bits respectively? What does this accomplish in the content of data parsing?

int32_t v = static_cast<int32_t>((static_cast<uint32_t>(* (payload.first + 17))) << 24) +:

The line declares a 32-bit integer and initializes it. Afterwards, it extracts the byte at the position 17 bytes from the start of payload.first. Furthermore, it casts it to an unsigned 32-bit integer, and then shifts it left by 24 bits.

(static_cast<uint32_t>(* (payload.first + 18)) << 16) +:

This line is similar to the previous operation. However, it is for the byte at position 18, which shifts it left by 16 bits to position it as the second highest byte.

(static_cast<uint32_t>(* (payload.first + 19)) << 8) +:

The line takes the byte at position 19 which shifts it left by 8 bits. Therefore, it makes it the second lowest byte in the resulting 32-bit integer.

static_cast<uint32_t>(* (payload.first + 20));:

The line takes the byte at position 20 which casts it directly to an unsigned 32-bit integer without any shift.

return static_cast<double>(v) / 100.0;:

This line converts the assembled 32-bit integer “v” to a double type, and divides it by 100.0 to produce the final value. This line implies that the data extracted from the payload is being scaled down by a factor of 100.

The code block is used to parse and assemble a 32-bit integer from four consecutive bytes in a data payload, which represents the q-axis current from a VESC.

The bytes are shifted by 24, 16, and 8 bits respectively to align them into their correct positions in the 32-bit integer.

Because it ensures the bytes are correctly combined in order to reflect their actual value, which can be used for further calculations.

Question 5: Explain the purpose of declaring the current_qaxis()method in the vesc_packet.h file. How does this relate to the changes made in the .cpp file? How does the declaration enable the VESC driver to publish this data?

The purpose of declaring the “current_qaxis()” method in the “vesc_packet.h” file is to provide an interface for accessing the q-axis current data from the VESC.

In the ".cpp" file, we add the "current_qaxis()" function to calculate the q-axis current value based on the data received from the VESC. However, simply implementing this function in a ".cpp" file does not make it accessible to other parts of the program. By declaring the "current_qaxis()" method in the "vesc_packet.h" file, the function becomes part of the VESC driver interface, which means that other parts of the program containing "vesc_packet.h" can call "current_qaxis()" to get the current data of the q-axis. Therefore, the driver can successfully publish the q-axis current by using the current_qaxis() method during the subscription.

Question 6: What is the purpose of adding state_msg->state.current_q = values->current_qaxis(); to the vesc_driver.cpp file?

The purpose of adding state_msg->state.current_q = values->current_qaxis(); to the vesc_driver.cpp file is to update the ‘state_msg’ object to contain the current value of the q-axis current from VESC. The q-axis refers to the axis in the FOC of the brushless DC motor. This line of code ensures that the real-time value of the current along the q-axis of the motor's magnetic field is included in the `state_msg` message, which is then published to other nodes in the ROS network.

Question 7: How does the addition of current_q to the VescStateStamped message affect the information that the VESC node publishes?

After adding "current_q" to the "VescStateStamped" message, the information published by the VESC node contains data about the motor q-axis current. This addition means that the VescStateStamped message can carry real-time information about the motor's q-axis current, which is a key parameter for controlling brushless DC motors.

Question 8: What is the purpose of modifying the arg name="port" line in the vesc_driver_node.launch file?

The purpose of modifying the arg name="port" line in the vesc_driver_node.launch file is to specify the communication port that VESC connects to the Jetson Nano or the computer. This modification ensures that ROS knows exactly where VESC is sending and receiving data.

Objective 3: Experiment with VESC using ROS

Question 9: Explain the role of each component in the command $ rostopic pub -1 /commands/motor/speed std_msgs/Float64 -- "7500.0". What does each part mean?

rostopic is a command line tool provided by ROS that allows us to publish, subscribe, list and find information on these topics.

pub is a subcommand of "rostopic", which stands for "publish".

"-1" tells the rostopic pub command to publish only one message and then terminate automatically.

/commands/motor/speed is the path of the topic to which the message will be published. In this case, it is a command intended to control the speed of the motor, since the VESC node subscribes to the speed topic.

std_msgs/Float64 -- "7500" specifies the message type being published, which contains a single 64-bit floating point number and sets the motor speed value to 7500rpm.

In summary, the entire command is used to publish a message containing the value 7500.0 to a topic named /commands/motor/speed in the ROS environment to which the VESC node is subscribed to.

Objective 4: Controlling VESC with the Joystick

Question 10: Explain why is editing the CMakeLists.txt file and adding the catkin_install_python command an important step?

The "catkin_install_python" command is used to install the Python executable node in the ROS package. Adding this command allows us to specify Python scripts as executables and where they should be installed in ROS. By editing this file and adding the "catkin_install_python" command, we can ensure Catkin understands the Python nodes and processes them correctly.

Question 11: Explain the code in the controlled.launch file. In your explanation, include the following. Explain what is the role of the Multiplier parameter in JoyControl.py. What changes occur when the multiplier is increased or decreased and how does it affect the duty cycle range? What is the controlMethod value and how does its value affect the AEV?

The "controlled.launch" file involves setting up the ROS startup file and using the joystick to control the vehicle. The Multiplier parameter in "JoyControl.py" is used to scale the input received from the joystick. This scaling affects the size of commands sent to the vehicle. When we control an AEV, this could adjust the speed or rate of turn.

Increasing the Multiplier will increase the range of commands sent to the vehicle, which will result in the vehicle accelerating faster or having more power. Conversely, reducing the Multiplier reduces the range, resulting in gentler movement. Duty cycle is the ratio of the active signal time to the total time considered and is used with pulse width modulation to control motor speed. Changing the Multiplier affects the pulse width and hence the duty cycle, which directly affects the speed of the motor. Increasing the Multiplier increases the duty cycle range achievable with joystick movement, allowing for faster movements. However, reducing the Multiplier has the opposite effect. It reduces the maximum duty cycle achievable at the joystick position. Therefore, the vehicle will make a gentler movement at a lower speed.

In the "controlled.launch" file, the controlMethod value is set to 0. When "controlMethod" is set to 0, it means that the vehicle will be in a state that requires manual control by the operator, usually via a joystick or other device.

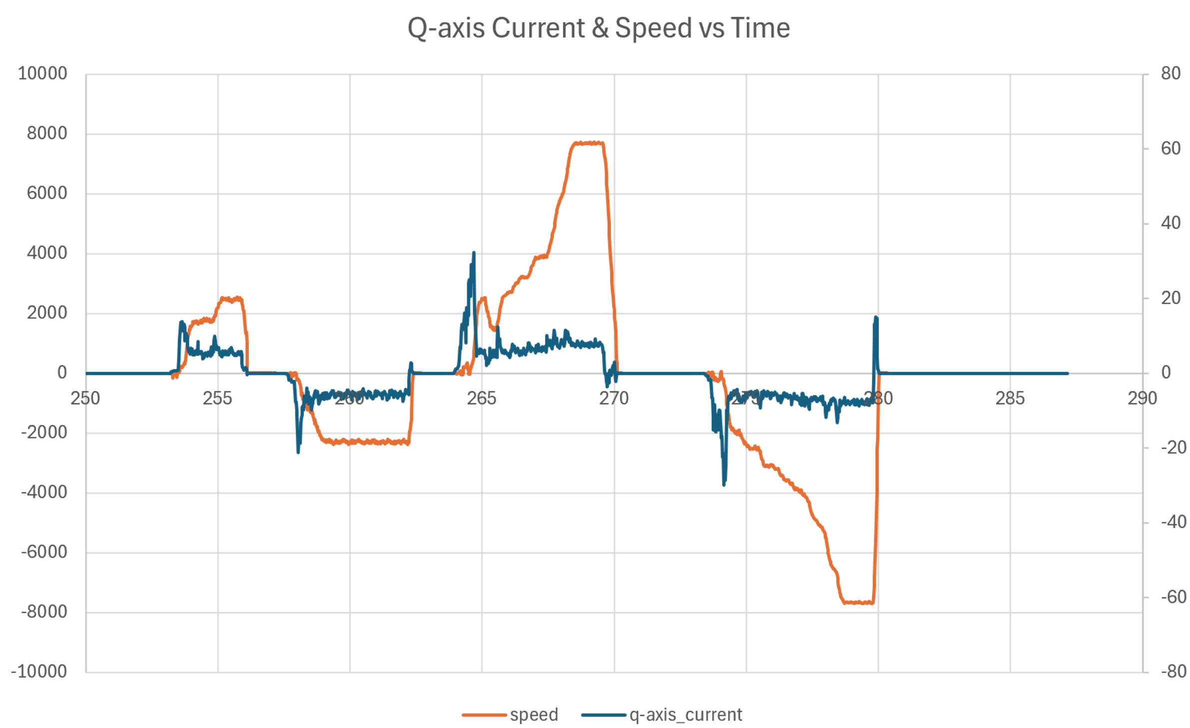
Question 12: What value does the line a = -1*data.axes[0]*0.5+0.5 have within the callback function of servoControl.py? Consider the effect on the direction of the servo motor.

a = -1*data.axes[0]*0.5+0.5

The data.axes[0] is the value from the joystick's first axis, its value is usually between -1 and 1 [-1,1]. -1*data.axes[0] reverses axis direction with multiply -1, the servo will move in the opposite direction to joystick movement. -1*data.axes[0]*0.5, this reduces the inversion value by a factor of 0.5. The range is between -0.5 and 0.5 now [-0.5, 0.5]. At last, -1*data.axes[0]*0.5+0.5, the range moves from [-0.5, 0.5] to [0, 1] by adding 0.5. Therefore, the value a is between 0 and 1 range [0,1].

Objective 5: Logging Data from VESC

Question 13: Attach your graph plotted in Microsoft Excel in your final report and make sure you have your axis labeled for time, speed, and q axis current. Explain your observations from the graph in detail. Comment on the relation between the speed and the q-axis current and explain your understanding thoroughly in at least 200 words. Provide examples from the lecture material on the operating principles of the motor to explain your answer.



According to the graph, the q-axis current increases quickly when the electric vehicle accelerates from rest. The current reduces when it achieves its top speed and begins to slow down.Then, the current continues to decrease as the analog stick is returned to it’s original position and then pushed backwards, the forward speed drops to zero and starts accelerating backward, which makes the current reach a negative maximum. As a result, the electric vehicle accelerates forward when a positive q-axis current is applied and backward when a negative q-axis current is applied.

The q-axis is the axis that creates torque, according to lecture materials on the principles of operation of electric motors. The q-axis current is proportional to the electromagnetic torque produced by the motor when it is operating. This is due to the interaction between the rotor's magnetic field and the q-axis current. Consequently, the torque production of the motor will be immediately impacted by any changes in the q-axis current. When the load is balanced, the motor will accelerate due to an increase in electromagnetic force and torque caused by an increase in q-axis current. On the other hand, if the motor speed is more than needed, it may be lowered by lowering the q-axis current, which will also lower the torque.