Real-time pendulum mimicking system using slit time-PWM-setpoint position lookup table and quarter period analysis

Chew Lik Siang
29035058
ECE4075 – Real time embedded systems
Electrical and Computer Systems Engineering
Monash University Malaysia
lche0045@student.monash.edu

Abstract

This project focused in developing an algorithm for mimicking the motion of a free-swinging pendulum based on the signal coming from the photoelectric sensor. The marker attached to the motor was able to correlate with the direction of the pendulum motion and follows the pendulum swinging speed as well as the swinging amplitude.

Keywords— Pendulum, Real time, Embedded, PID

I. Introduction

In this project, this pendulum motion mimicking task is treated as a real time ongoing process for the system to monitor. Since real time embedded systems are relatively small and power limited, the algorithm that needed to perform on the board should be able to optimize the limited memory and execution power. This system will utilize both software and hardware methods to capture the information of the pendulum motion from the photoelectric sensor and reproduce the same motion onto the DC motor attached with a marker.

Firstly, understanding the properties of the mechanical system to mimic is vital for our design of algorithm. A conventional pendulum system can be viewed as a point mass connected to the end of one string. Under normal circumstances, if the point mass is released from a position away from its equilibrium point, a repeating oscillation can be observed till the pendulum ultimately stop at the initial equilibrium point due to the losses of air resistance and heat.

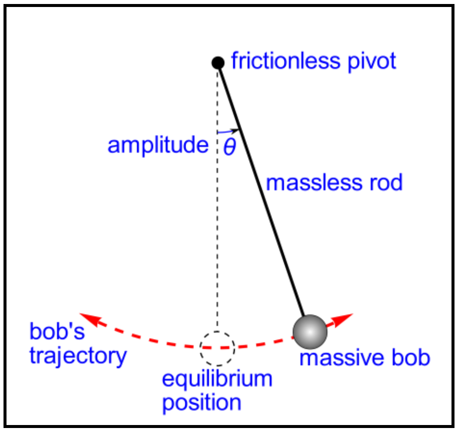

Figure 1: Conceptual drawings of a simple pendulum system [1]

With the small angle approximation (smaller than 0.1 radian), the period of the pendulum can be modelled as follows [2],

$$T = 2\Pi\sqrt{\frac{L}{g}}$$

T = pendulum period
L = length of the string
G = gravitational field strength

In this equation, the period is claimed to be independent of the angular position of the pendulum.

In order to account for the decaying property of the non-ideal pendulum system, the amplitude of a pendulum can be modeled using the following equation.

$$Amplitude = Ae^{\frac{-b}{2m}t}$$

A = Initial amplitude
b = damping constant of the system
m = moment of inertia of the system

With this, the overall motion of the pendulum system can be modelled using this equation,

$$y(t) = Amplitude * cos(2\pi t \,/\, T)$$

T = period
t = time at the point of interest

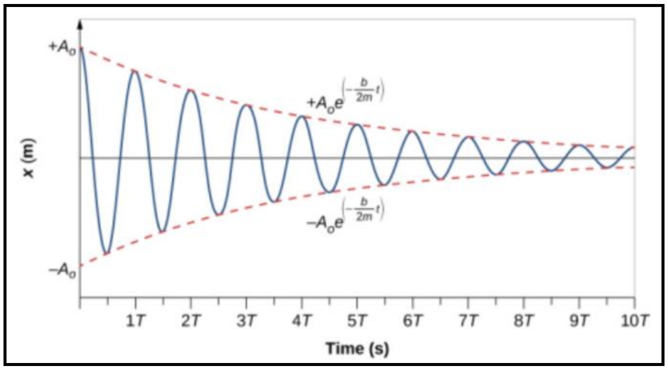

Figure 2: Position versus time for the mass oscillating on a pendulum. Notice that the curve appears to be a cosine function inside an exponential envelope [3].

In this project, the pendulum motion will be mimicked in a real time manner with the interruption time from the photoelectric sensor as the only source of information. The speed of the motor as well as the desired angular position for the motor to reach will be computed according to the interruption time collected by the photoelectric sensor. The relationship between the interruption time, PWM and desired position will be governed by a "slit time-PWM-position" lookup table.

The following learning objectives will be achieved upon completion of this project:

1. Develop and practice real time embedded system design, testing and debugging techniques.

2. Understand the trade-off between hardware and software implementations in a real time embedded system by studying a software hardware co-design problem.
3. Interpret timing information from a sensor and its relationship to a physical system.
4. Design modules that respond to sensor changes within hard deadlines.
5. Design a PID motor controller with PWM output to produce desired motions in real time

II. DESIGN

a) A technical drawing of the pendulum mimicking structure.

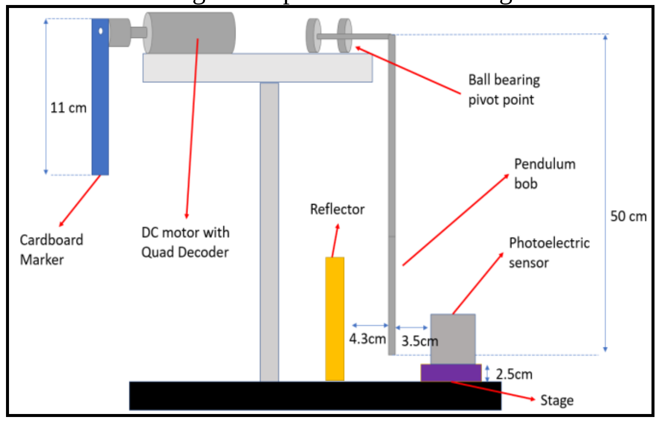

Figure 3: Overall structure of the mechanical setup with measurements

As shown in Figure 3, the overall pendulum structure is represented by a metal plate with an aperture connected to a free-swinging pendulum pole. The pendulum bob is covered by a black tape in order to better absorb the wave shoots out from the photoelectric sensor whenever the pendulum crosses the photoelectric sensor region. At another side, a marker is connected to the DC motor which equipped with a quadrature decoder. This marker will be the indicator of the motor movement to shows the performance of the system in mimicking the actual pendulum motion.

The signals from the encoder and the photoelectric sensor will be fed into the Arduino Due board. The board which connected to a 12 V supply will then be used to power up the photoelectric sensor, motor, and encoder.

b) Circuit schematics.

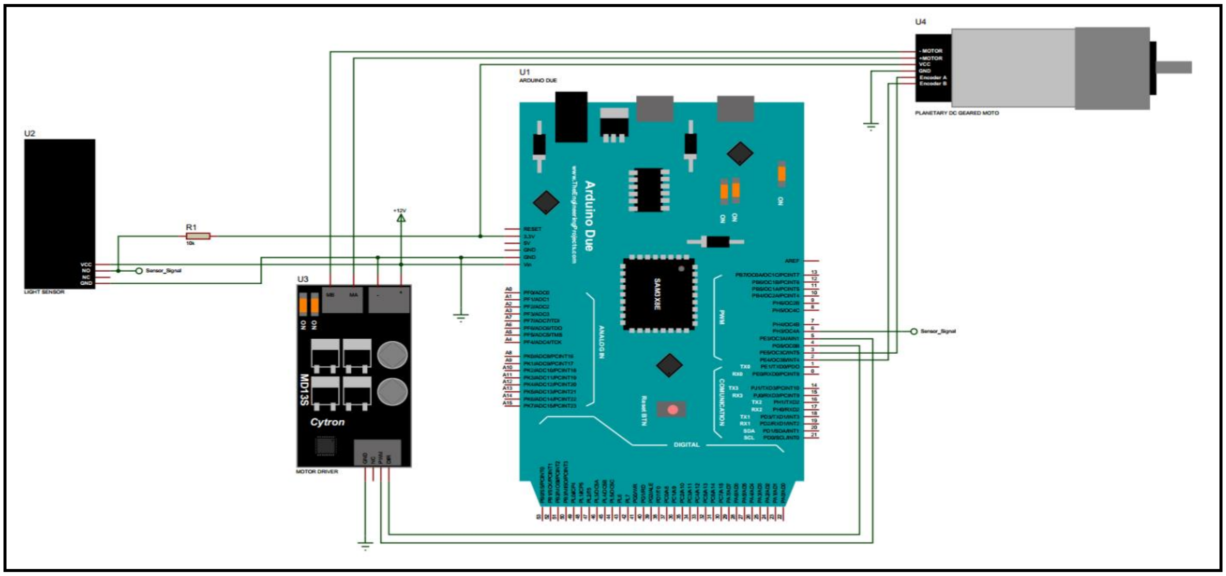

Figure 4: The circuit schematics (connected to DC motor, Motor Driver, Photoelectric Sensor and Arduino Due Board)

c) Technical drawing describing the method of identifying the direction and speed of the pendulum.

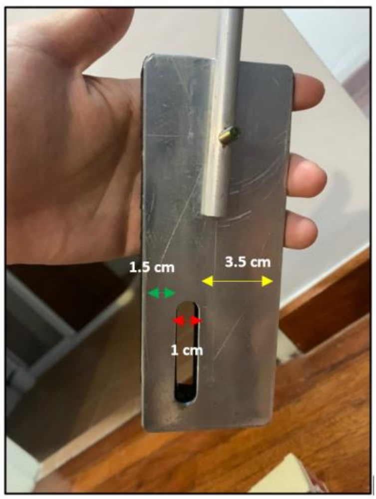

Figure 5: Measurement of the pendulum blob and the aperture

Based on Figure 5, it is clear that the aperture at the pendulum bob was deliberately located slightly away from the center position. Due to this asymmetricity, the duration of the photoelectric sensor getting blocked when the pendulum passes from left and right will be different. This duration can be picked up by the algorithm by making the photoelectric sensor to be both rising and falling edge triggered and hence, referring to Figure 6 and Figure 7, the sequence of the T_short and T_long will be different according to the direction of the pendulum.

The duration of the T_short will be shorter if the pendulum bob moves at a higher angular speed and hence resulting in a shorter period of time the photoelectric sensor getting blocked. In this regard, based on the magnitude of the T_short collected, the angular speed of the pendulum bob can also be estimated accordingly.

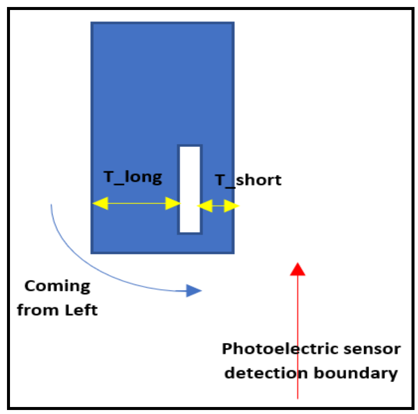

Figure 6: T_short first then followed by T_long detected by photoelectric sensor (Coming from the Left)

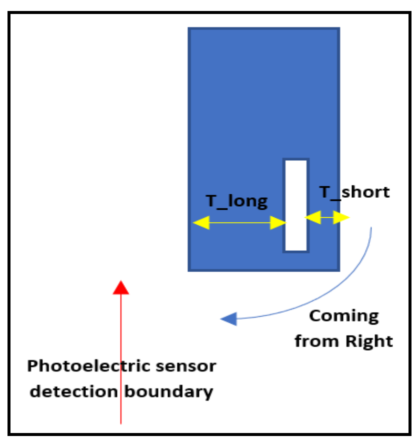

Figure 7: “T_long” first then followed by “T_short” detected by photoelectric sensor (Coming from the Right)

d) Flowchart describing the mimicking algorithm.

A. Half period and quarter period explained.

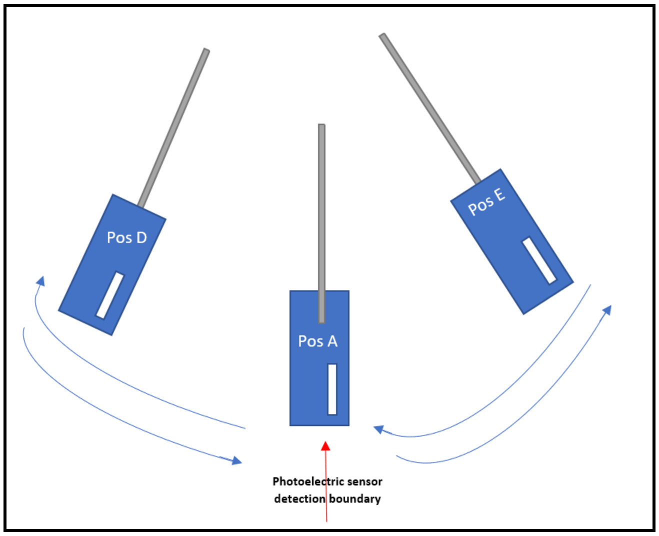

Figure 8: Break down of the stages to complete 1 oscillation of the pendulum.

In Figure 8, a complete cycle of pendulum motion can be described as having the pendulum bob moving from Pos E → Pos A → Pos D → Pos A → Pos E. With this understanding, the duration for the pendulum to complete one cycle can be broken into 4 durations as indicated by 4 arrows in the Figure above. Assuming the pendulum is crossing the photoelectric sensor boundary (Pos A) from the right (Pos E), the second time the photoelectric sensor getting triggered will be the duration for the pendulum to move from Pos A to the maximum amplitude (Pos D) and then back to Pos A again. This duration will be addressed as half duration in this project. It is worth noticing that the duration of the pendulum bob staying at the maximum amplitude will be roughly the same as the quarter period that can be easily derived from the half period.

B. Flowcharts of the pendulum motion mimicking algorithm

The general operation stages of the pendulum motion mimicking system are as shown in the Figure 9.

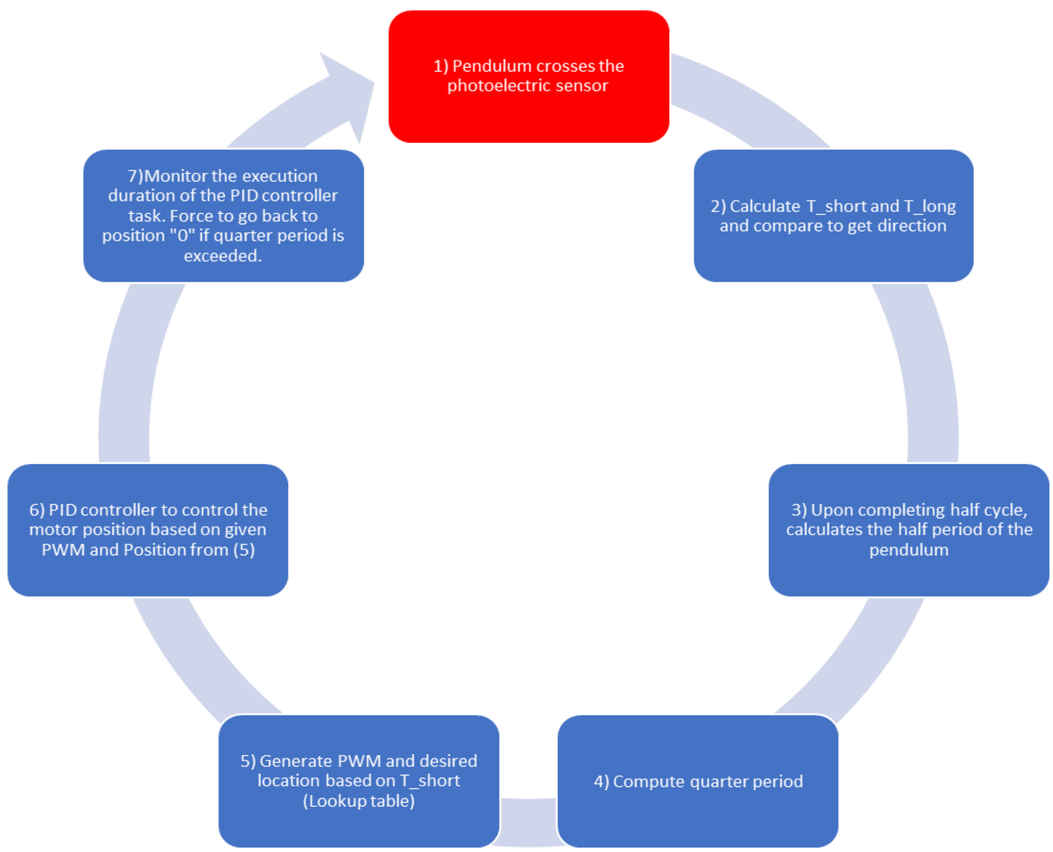

Figure 9: General workflow of the pendulum motion mimicking system

At the first stage, the photoelectric sensor triggers an Interrupt Service Routine (ISR) whenever a rising edge or falling edge is detected. In the ISR, the following information will be computed depending on the stages in the ISR:

Table 1: Summary table of the info to be computed by the ISR at different stages of the process.

Info to be computed	Stage
Interruption period (T_short and T_long)	The pendulum bob is currently passing through the sensor region (Pos A)
Direction of the pendulum motion (by comparing T_short and T_long)	Once the pendulum blob completely passed through sensor boundary
Half period of the pendulum	Once pendulum bob passed through sensor region twice. Assuming pendulum motion to be going to left and then to the right, the half period = time_at_2 – time_at_1

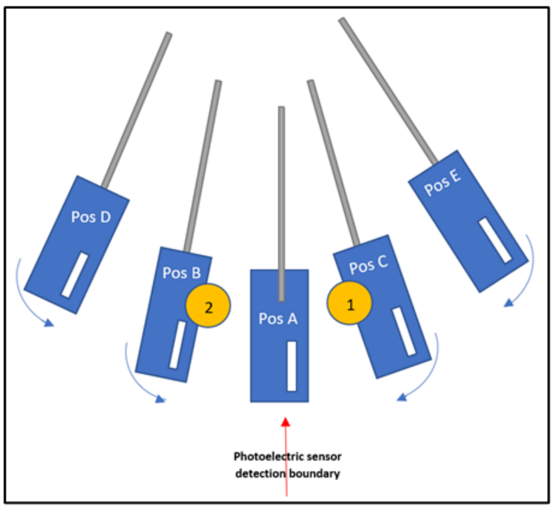

Figure 10: Representing the motion of the free-swinging pendulum with timing at (1) and (2) marked for calculation of half period

Since the half period duration is essential to determine the duration of the pendulum bob staying at the maximum amplitude, the pendulum mimicking system will only start to match the motor motion with the pendulum at the second rising edge of interruption by the blob. At this point, both the marker and the pendulum are at the equilibrium position and should be roughly aligned to each other. This is the starting point of the pendulum mimicking process.
Upon starting the mimicking process, every beam interruption of the photoelectric sensor will give the information of the current pendulum motion as indicated in Table 1. This information will be used to compute the direction of the motor, the setpoint position (in term of timer counter value), as well as the driving PWM magnitude. Once the motor monitoring task is initiated, it will start to monitor its execution time and upon reaching the quarter period timing, the setpoint position will be automatically set to 0 to mimic the falling motion of the pendulum as explained in Figure 8.
The setpoint position (in term of tc value) and the motor driving PWM are computed by referring to a manually constructed look up table shown below according to the period photoelectric sensor getting blocked, "slit_time".

Table 2: Slit time-PWM-setpoint position lookup table

Slit_time (Ticks)	**Set point position** (magnitude) *in terms of timer counter value*	**PWM** (Duty cycle)
5	55	18
7	51	18
8	50	18
9	33	18
10	33	17
11	33	17
12	30	17
13	30	15
14	30	15
15	30	15
16	30	11
17	28	11
18	28	11
19	23	11
20	23	11
21	23	10
22	20	10
23	20	10
24	20	10
25	20	9
26	20	9

27	20	9
28	15	9
29	15	9
30	15	8
32	15	8
33	15	8
34	15	8
35	15	7
36	15	7
39	13	7
40	13	7
42	13	7
44	13	7
47	13	7
48	13	7
53	13	7

The range of the available positions to the motor are within 13 to 53 timer counter value. The exact position with the direction can be easily computed by using the following equation,

$$desired_{position} = direction * magnitude\ of\ position\ from\ lookup\ table$$

which direction is expressed in term of +1 for clockwise and -1 for anticlockwise.

This computed setpoint position will be passed into the PID controller as the target with respect to the current motor position. The output from the PID controller can be summarized as follows,

$$output\ from\ PID = \begin{cases} < 0, & if\ the\ current\ position\ is\ bigger\ than\ set\ point\ position \\ \approx 0, & if\ current\ position\ is\ close\ to\ set\ point\ position \\ > 0, & if\ the\ current\ position\ is\ smaller\ than\ set\ point\ position \end{cases}$$

The mimicking algorithm is explained in detail in Figure 11 to Figure 13.

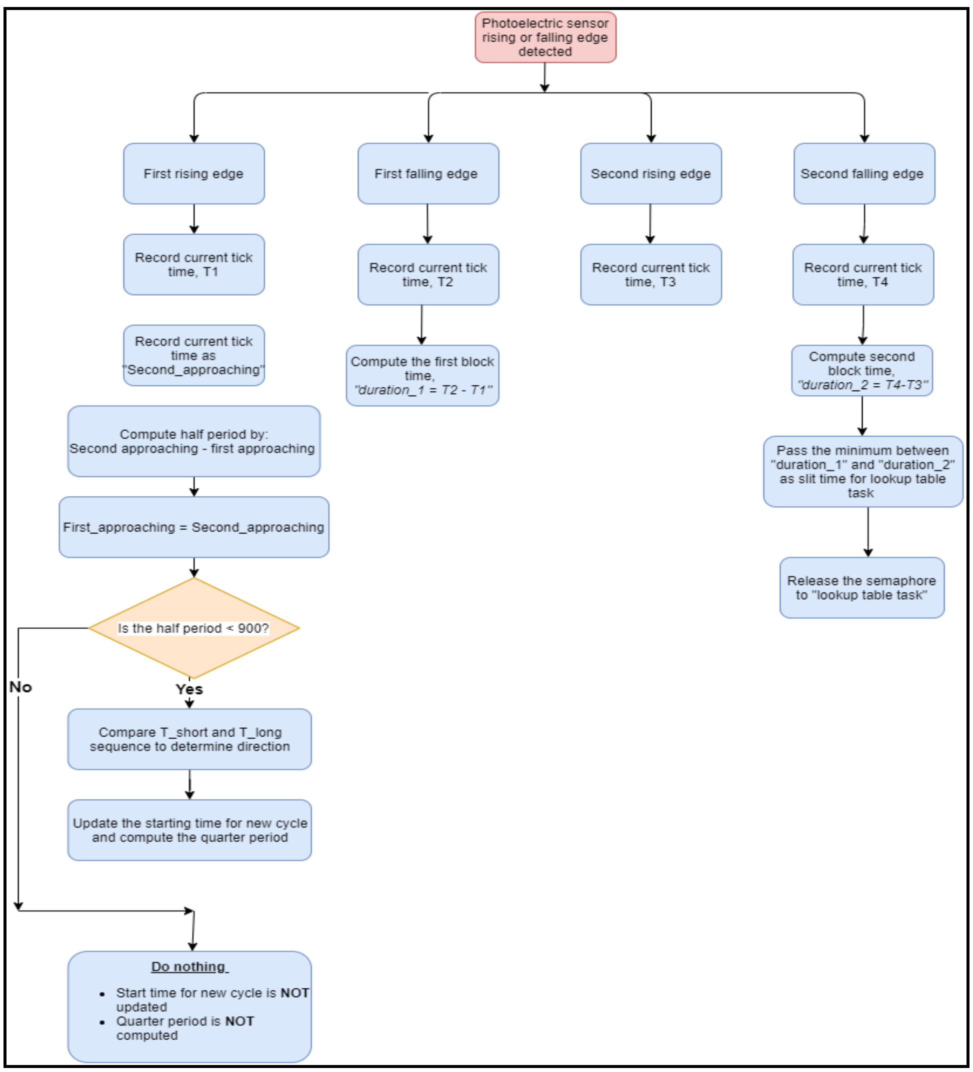

Figure 11: The code flow of Photoelectric sensor ISR

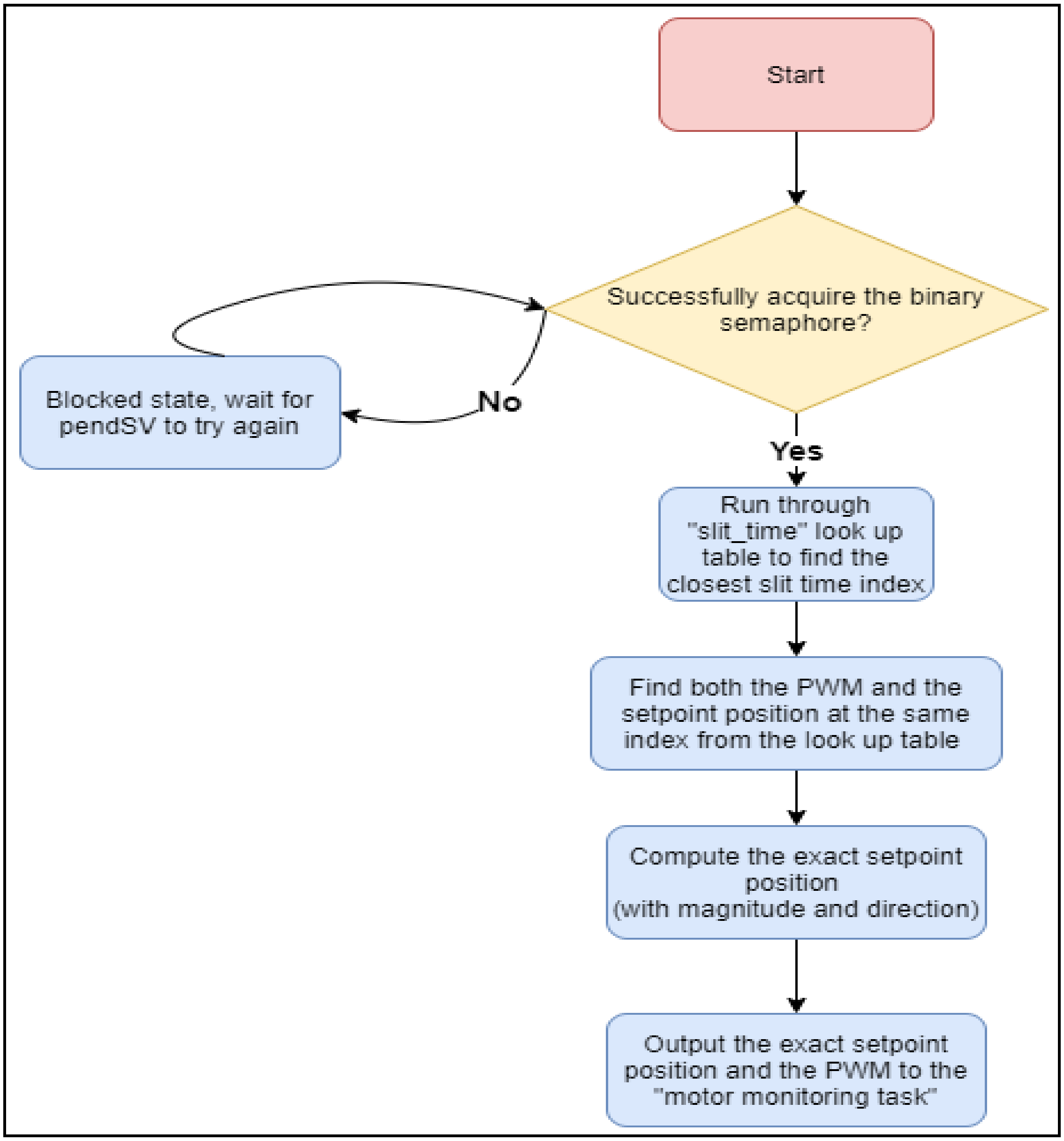

Figure 12: The code flow of "Look_up_table" task (highest priority task)

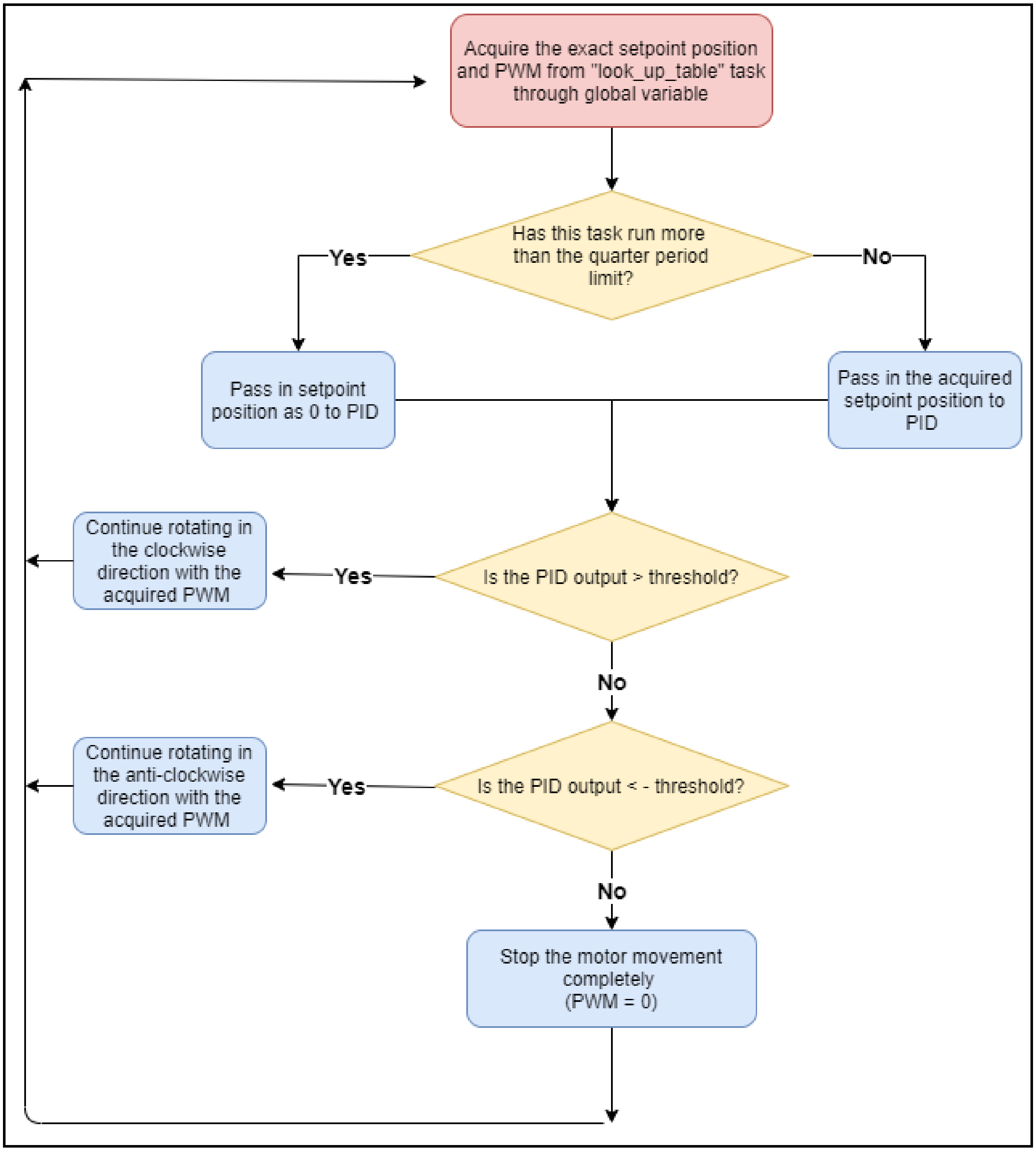

Figure 13: The code flow of "Motor Monitoring" task (lowest priority task)

III. Results & Discussion

A. *Checking for the relationship between the starting position of the swinging pendulum and the starting direction*

1) *Proposed methods:*

The pendulum bob was released from a predetermined angle ranging from 75 degrees to 30 degrees with 15 degrees interval. Once the marker of the motor begins to move, the initial rotation direction was recorded and used to evaluate the direction tracking capability.

2) *Results:*

Table 3: Summary table of the direction synchronization of the system

	Pendulum direction	
Release angle (°)	Clockwise	Anti-clockwise

30	Correct initial rotating direction	Correct initial rotating direction
45	Correct initial rotating direction	Correct initial rotating direction
60	Correct initial rotating direction	Correct initial rotating direction
75	Correct initial rotating direction	Correct initial rotating direction

3) Inference from the results:

The motor with the marker is capable of starting the mimicking process in spinning in a correct direction with reference to the swinging pendulum motion regardless of the starting angle and the starting direction.

However, it is worth noticing that the mimicking process only begin after 1 complete oscillation of the pendulum. With the result shown in Table 3, the algorithm that analyse the sequence of the “T_short” and “T_long” during the photoelectric sensor interference period to predict the swinging direction is reliable.

B. Checking the effect of manipulating the Kp,Ki and Kd values.

1) Proposed methods:

The PID controller in this project is used to monitor the exact position of the motor instead of the PWM value. The pendulum bob was released from 90° angular position and allowed to swing freely until it finally stops at the equilibrium position. The overall performance in term of the capability of matching the resultant angular position of the system is evaluated under different combination of Kp, Ki and Kd values. The Kp value will be fixed at 2 for simplicity and only Ki and Kd values will be manipulated.

2) Results:

Table 4: The observation with regards with different combination of Kp,Ki and Kd

PID parameters (Kp, Ki, Kd)	Observation
Kp = 2, Ki = 0.5, Kd = 0	Motor marker overshoots the desired angular position by a considerable amount whenever the pendulum begins to change direction. Occasionally, the system becomes totally unstable after a few cycles.
Kp = 2, Ki = 0.0001, Kd = 0	Motor marker could not reach the desired angular position in time and begin to lag over whenever the pendulum begins to change direction. The system remained stable throughout the mimicking process.
Kp = 2, Ki = 0.001, Kd = 1	Motor marker able to reach the desired angular position with a small amount of delay as compared to the actual pendulum. The system remained stable throughout the mimicking process.

3) Inference from the results:

Generally, a big Ki values increase the ability of the system to react to input faster. However, with such a big Ki value as shown in case 1, Ki = 0.5, the systems became unstable and overshoot the desired position for a considerable amount. The effect of Ki is further proven by the combination in case 2, which the Ki value is much smaller, Ki = 0.000. This causes the system to require much longer time to react to the new input.

Finally, for the final model, “Kd” term is introduced, and this definitely helped to enhance the stability of the system as the derivative of the error between the desired set point position and current position is taken into account. The final combination of “Kp = 2, Ki = 0.001, Kd = 1” has shown a reasonable performance in term of matching the desired angular position of the motor and has been implemented in the final prototype.

C. Checking if the marker rising and falling speed matches with the swinging pendulum angle

1) Proposed methods:

The pendulum bob was released from a predetermined angle to initiate the pendulum mimicking process. After 2 complete oscillations of the pendulum, the recorded motor marker motion is replayed in slow motion to perform a rough evaluation of the smoothness of the mimicking system.

Since in this pendulum mimicking algorithm, the only controlled parameter by the PID controller is the exact position that need to be reached by the motor. However, how fast the motor achieves the desired angular position is not controlled by the PID controller. If the motor reaches the desired position too early as compared to the actual swinging motion, a temporary pausing motion might be observed at the maximum amplitude and reduces the overall smoothness of the mimicking process.

2) Results:

Table 5: Observation of the synchronization ability of the system for different release angle

	Pendulum direction	
Release angle (°)	Clockwise	Anti-clockwise
30	An observable small pausing motion detected at the maximum amplitude.	An observable small pausing motion detected at the maximum amplitude.
45	Similar to the observation for 30°	Similar to the observation for 30°
60	Smooth motion with almost no pausing motion detected at the maximum amplitude.	Smooth motion with almost no pausing motion detected at the maximum amplitude.
75	Similar to the observation for 60°	Similar to the observation for 60°

3) Inference from the results:

The algorithm was able to mimic the swinging pendulum motion flawlessly at slightly bigger release angle scenarios. At big angular position (60°-75°), a constant PWM to drive the marker to the computed position can well approximate the actual decaying speed motion of the swinging pendulum.

However, due to the non-linearity of the actual swinging pendulum motion, the oscillatory characteristics between the distance travelled and the angular velocity of the pendulum at small angular position changes much rapidly. This causes our algorithm which uses a precomputed PWM to drive the motor to the desired position to sometimes reach the position slightly earlier and hence a pausing motion was detected.

In order to approximate the non-linearity of the free-swinging pendulum motion, a few methods have been implemented.

1. The motor will return to the origin upon exceeding the computed quarter period.
2. A lower PWM is used when the desired position for the motor to reach is small (30°-45°)

D. *Checking the tolerance towards unexpected disturbance (pendulum held in mid air)*

1) Proposed methods:

The pendulum was first released from 90° angular position to provide enough energy into the system and allowed to oscillate freely for few cycles. Afterwards, the pendulum will be held in mid-air before releasing it to a free-swinging motion once again. The system mimicking ability to recover from such disturbance is examined.

2) Results:

At the initial stage when there is no interruption introduced to the system, the system can mimic the pendulum motion perfectly. When the pendulum is blocked from going back to the equilibrium position, the motor marker completes its current cycle by going to the appropriate angular position computed based on the previous slit time before stopping at the equilibrium position. A maximum 10° occasional deviation from the equilibrium position is observed due to the threshold allowance set in the PID controller.

Once the pendulum is released again from the blocked position, the motor will only begin to mimic the pendulum motion again at the second interruption of the photoelectric sensor.

3) Inference from the results:

It is worth noticing that, when the pendulum first released and allowed to pass through the photoelectric sensor region for the first time after the interruption, the motor will not match its motion. This is due to the fact that due to the previous interruption of holding the pendulum, the half period computed is unreasonably high and has been ignored, hence the algorithm will need to recompute a half period at the second interruption of signal and begin the mimicking motion once again.

E. *Checking the tolerance towards unexpected disturbance (External force applied to swinging pendulum during the free swinging process)*

1) Proposed methods:

The pendulum was first released from 90° angular position to provide enough energy into the system and allowed to oscillate freely for few cycles. The pendulum will be interrupted by exerting extra force whenever the pendulum reaches the maximum amplitude and about to switch its direction. The system mimicking ability to recover from such disturbance is examined.

Since the exact magnitude of force exerted is hard to obtain, the magnitude of the force exerted into the system is tabulated as the resultant angular angle of the pendulum as shown in table.

2) Results:

Table 6: Summary of the system's capability to handle external force interrupt of different magnitudes.

Maximum angular angle after the exertion of extra force	Observation
30° - 45°	The marker was able to mimic the pendulum well in term of angular speed and resultant angular position, with only minor deviations and sudden increase in motor rotational speed. Marker was able to recover from interrupts without issue.
45°-60°	Similar to above.
60°-75°	Similar to above.
>75°	The marker was able to mimic the pendulum motion however, the maximum amplitude reached by the motor is slightly lesser than the one travelled by the pendulum. After a few cycles, the system will be able to catch up with pendulum motion once again in term of angular position and angular speed.

3) Inference from the results:

The results show that the algorithm that work based on the slit time obtained during the photoelectric sensor method to determine the respective angular position set point and PWM is able to cope with the abrupt interruption. In fact, as long as this external force interruption does not happen during the photoelectric sensor interruption period (close to the photoelectric sensor boundary region), this interruption will still be able to be captured by the photoelectric sensor when it passes through it.

However, when the external force results in more than 75° resultant angular position, the motor will not be able to match with the big amplitude position of the pendulum. This is due to the fact that, the “slit time-PWM-position” lookup table is discrete and has a finite resolution. The lookup table was made by manually tabulating the slit time recorded at the photoelectric sensor when the pendulum is released at a certain angle. Since >75° is not considered when the lookup table was built, the algorithm will not be able to find the appropriate angular position to go to.

IV. Conclusion

In conclusion, with the implementations of well-designed software algorithm and hardware installation, the marker controlled by the motor is able to mimic the free-swinging pendulum’s motion with only a slight deviation. The quarter period algorithm and also the position controlled PID allow the system to be able to mimic the pendulum motion with a high accuracy with minimal or no overshoots. A drawback of this system is that due to the finite resolution of the lookup table, any unpredicted motion such as having a gigantic swinging amplitude or very high swinging speed will result in a big deviation in the mimicking system. Future work that will be considered includes developing a continuous equation that relates the slit time, the motor PWM and the desired set point position for the motor to go to. This will eliminate the needs of developing a lookup table manually which suffers from limited resolution as well as wasting the unnecessary effort of taking readings. In regard of computational time, computing the parameters through a single equation will definitely reduce the computational load resulting in a much smaller latency between the update of a new PWM and the setpoint position.

References

[1] "Pendulum - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Pendulum. [Accessed: 30- May- 2021].

[2] [1] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of physics*. New York: Wiley (in English), 1997.

[3]"15.6: Damped Oscillations", Physics LibreTexts, 2020. [Online]. Available:https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_University_Physics_(OpenStax)/Book%3A_University_Physics_I_-_Mechanics_Sound_Oscillations_and_Waves_(OpenStax)/15%3A_Oscillations/15.06%3A_Damped_Oscillations. [Accessed: 30- May- 2021].

V. APPENDIX

Images of the pendulum system from different view:

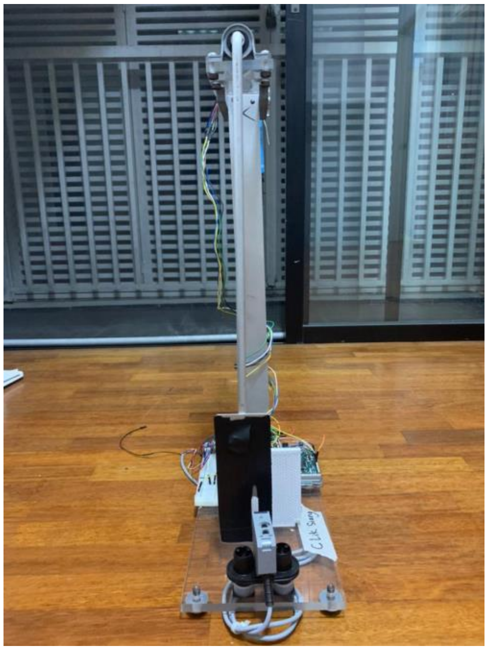

Figure 14: Front view

Figure 16: Rear view

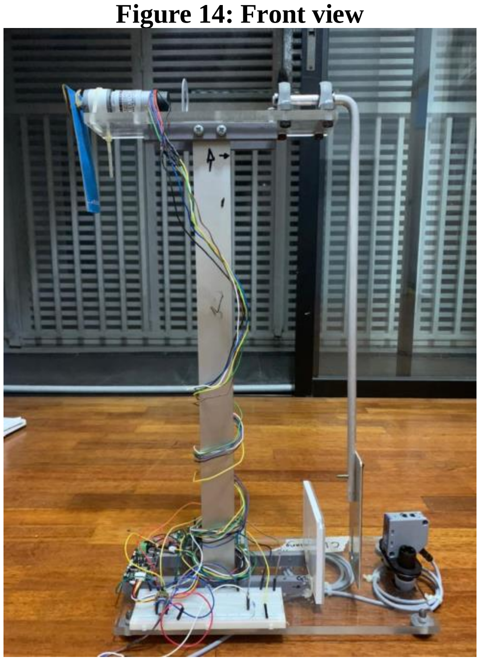

Figure 15: Left side view

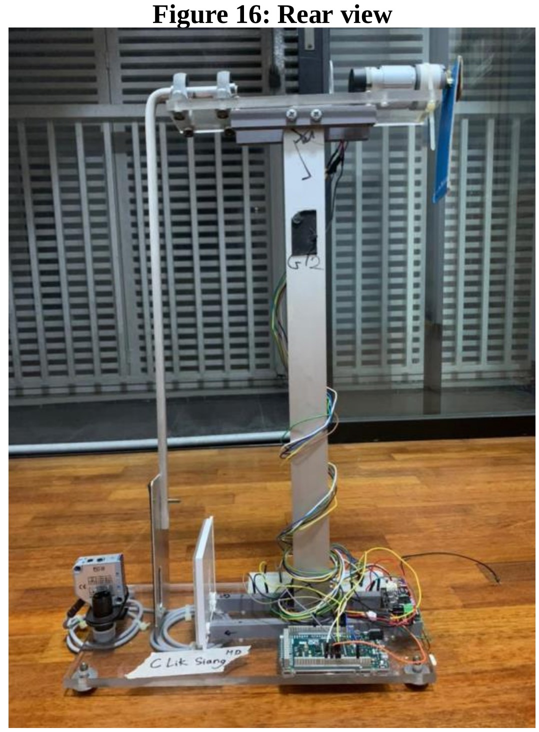

Figure 17: Right side view