Name: Sim Justin

Matric No: A0257926N

Project 1: PID Control

# Initial Conditions

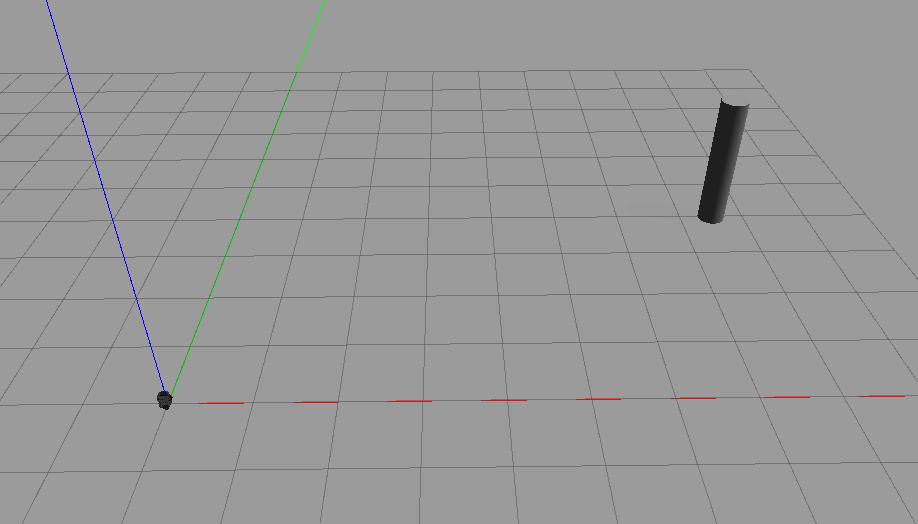


Figure I: Initial Location of Pillar and Turtlebot3

Starting Location of Pillar: (7.5, 4)

Stopping Distance: 1.1

Initial Distance: 8.5

Orientation of Pole: 0.489957326 rad

# Implementation of PID Control

Integral Term: Integral term was defined by principles of Riemann Sum, whereby the error integral is estimated by taking the summation of the product of the error at each point of time and its associated delta time. As opposed to a Classical Riemann Sum, Trapezoidal Riemann Sum was implemented to improve accuracy of integral estimation (Stephanie, n.d.) by taking the area of the trapezoid encompassed by current error, previous error, and their associated timestamps. To avoid the issue of integral windup, the integrator is only active while within a controllable region as defined by the Proportional Term.

Derivative Term: Derivative term was defined through concept of differentiation by first principles, taking the gradient between the current error and the previous error as an estimate of the gradient of the tangent of the curve at the current point.

PID Control Term: PID term was defined by taking the summation of the product between each of the proportional, integral and derivative terms and their associated constants.

Implementation can be seen as follows:

A computer screen shot of a program

Description automatically generated

Figure II: Implementation of PID Control Terms

Angular error regulation is done by normalising *error\_angle­* to the range of -PI to PI, by adding 2 PI to it if it is smaller than -PI or adding the same value if it is larger than PI, until it falls within the prescribed range. Implementation can be seen below.

A black background with white text

Description automatically generated

Figure III: Angular Error Regulation

Angular control signal is limited by imposing a limit to both the angular velocities such that if the control signal output by the PID controller exceeds this limit, the angular velocity is written to be the maximum permissible angular velocity instead.

Linear control signal limiting was performed similarly. Implementation for both can be found below.

A computer code with text

Description automatically generated

Figure IV: Control Signal Limitation for Angular and Linear Velocities

# Tuning of PID Control

## Tuning Process

Tuning of linear and angular velocities took place concurrently. The tuning process was done using the CHDW Method (University of Exeter Department of Physics and Astronomy, n.d.). This was achieved by determining the proportional gain first, finding the point at which the system just begins steady oscillations then noting the period of oscillation and reducing this gain by around a third. Following this, the derivative gain was determined by testing with a range of values for the gain based on a third to half of the period noted above, until it was found that the Turtlebot’s linear and angular motion was critically damped. Lastly, the integral gain was determined by raising it until it resulted in oscillation in the movement of the Turtlebot, then reduced by a factor of 3. Finally, each of the gains were then fine-tuned further until critical damping of movement of the Turtlebot was achieved.

## Controller Characterisation

Each of the three gains are responsible for different roles in enabling a controller to perform its role. The proportional term is key in helping a system reach its target setpoint without inducing significant time delays in the system, and forms the backbone of the controller. The integral term’s purpose is to eliminate steady state error, which is especially helpful in cases where there are constant external forces acting on the system, or if there target setpoint were to be a moving target instead. The derivate term takes the role of damping the system, improving its overall stability, and minimising overshoot, thereby reducing oscillations in the system.

In this case, it can be found that a P-controller would be able to guide its system towards the setpoint, which involves an overshoot followed by significant oscillations about this setpoint before settling. A PI-controller would be similar in its behaviour, with the exception that it would be able to eliminate steady state error, if any. A PD-controller differs as the damping offered by the derivative term allows for the elimination of overshoot, although it would be unable to resolve for steady state error due to the lack of the integral term, which may lead to an asymptote at the setpoint. A PID-controller combines the benefits of all three terms as mentioned above, forming a system which is reactive to change while ensuring that there is little to no overshoot and achieving no steady state error.

## Design Discussion

Integral and derivative gains were kept low to minimise settling time and overshoot, with care taken to make sure the lowest required value that satisfied the challenges requirements was used to prevent the introduction of the feedback loop over-responding to errors, especially at smaller error values. Notably, given the lack of a significant source of steady state error as part of the task, integral gains were maintained at a lower than expected value.

The following gains were implemented as part of the final design:

A screenshot of a computer

Description automatically generated

Figure V: PID Gains for Linear and Angular Velocity

# Performance of PID Control

## Results

A graph with a line

Description automatically generated

Chart I: Linear Error over Time

A graph with a line

Description automatically generated

Chart II: Angular Error over Time

It can be observed in both linear and angular errors that there is minimal overshoot before the system settles at its respective target setpoint (achieves zero error). Steady state error is also largely eliminated with both errors settling at the zero mark and stability is achieved with little oscillation about this setpoint. A trade-off has been made with slightly longer settling time as a result however, but considering the slower speeds of the Turtlebot, this was largely inconsequential seeing that a majority of the time the Turtlebot spends correcting itself is within the linear region, i.e. the PID control term exceeds the maximum permissible linear and/or angular velocity of the Turtlebot. This allows for a well-balanced solution to the task at hand.

# Conclusions

A well-tuned PID-controller can enable a system to reach its target setpoint effectively and efficiently. Despite having prior experience in implementing PID-controllers in other applications, this project has provided me with an avenue to learn more about what I had been doing in an official capacity, along with testing and trialling improvements that I had previously not have the time to attempt, such as the CHDW tuning method and the conditional integrator solution to integral windup, as opposed the more band-aid solutions I had made use of in my past projects. This project has given me a greater appreciation of the complexities of designing control systems and has driven my passion in this area.