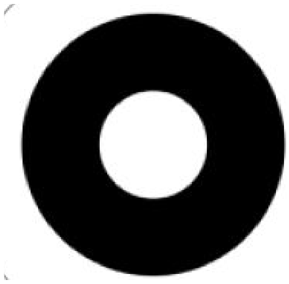
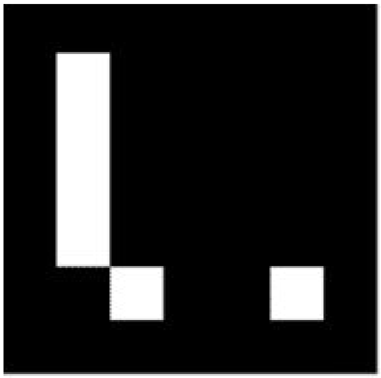
What are whycon markers and how do they work?



- Whycon markers are artificial markers used to perform localisation and pose estimation, especially on resource constrained systems. It was mainly developed for aerial and swarm robotics.
- It is computationally highly efficient, and it is easily deployable, it is available as an open-source ROS module.
- Whycon is basically a black and white roundel with known dimensions.
- **Working**
- Stage 1 : The algorithm employs a fast fill technique(**the advantage of this algorithm is that it can start from anywhere on the marker without performance penalty**) to search for touching segments of dark pixels(basically the outer layer - see the image). The ratio of each segment's pixels to its bounding box area is then compared to a value calculated from the known dimensions of the searched pattern.(Thresholding, if it is above a set threshold then the test will be passed), **the system now has a dark contiguous segment.**
- Stage 2 : Now with this segment's centroid the algorithm starts searching for another bright pixel(**white part)** and again it compares it with values generated from its bounding box dimensions(**Remember, the algorithm knows the dimensions of the marker).**
- If this stage 2 is passed then it means that the algorithm has found two elliptical,roughly concentric patterns.
- Once the segments found pass this test, now we have to calculate the **3D position,** of the marker. It is calculated by using its own dimensions, camera parameters and coordinate system(basically changing of bases vectors in a way by which we can estimate the pose of the robot i.e. by applying concepts of transforms).

What are ArUCo markers and how do they work?

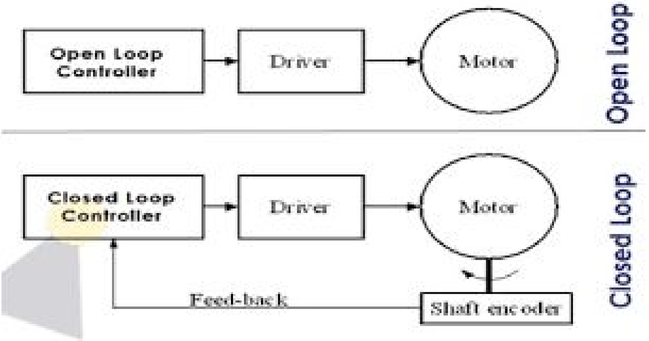


- An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id)
- The black border facilitates its fast detection in the image and the binary codification(**The white part**) allows its identification.
- Both Whycon and Aruco solves the problem of localisation, the only difference is the distance from which these markers can be made out by a camera.When using the same camera the maximum detection for ArUco marker was 4.4 m and for Whycon 13.181 m(Result of a research conducted by a team of researchers in TU Darmstadt).
- Each aruco marker has its ID that is distinguished by the distinct pattern of white strips inside.

- **Working.**
- It is a two stage process, **Stage 1:** The position of its four corners in the image (in their original order).**Stage2:** The identification of the marker.
- Stage 1: The contour of the image that resembles the marker is detected, if that contour does not match square it is discarded if it matches then stage 2 is performed.(This is done irrespective of the orientation of the marker.
- Stage 2: Each bit is extracted from the marker and it is compared with the dictionary to see if it matches any known binary codification.(Dictionary is the important part in detecting markers, it has a dictionary size and marker size).

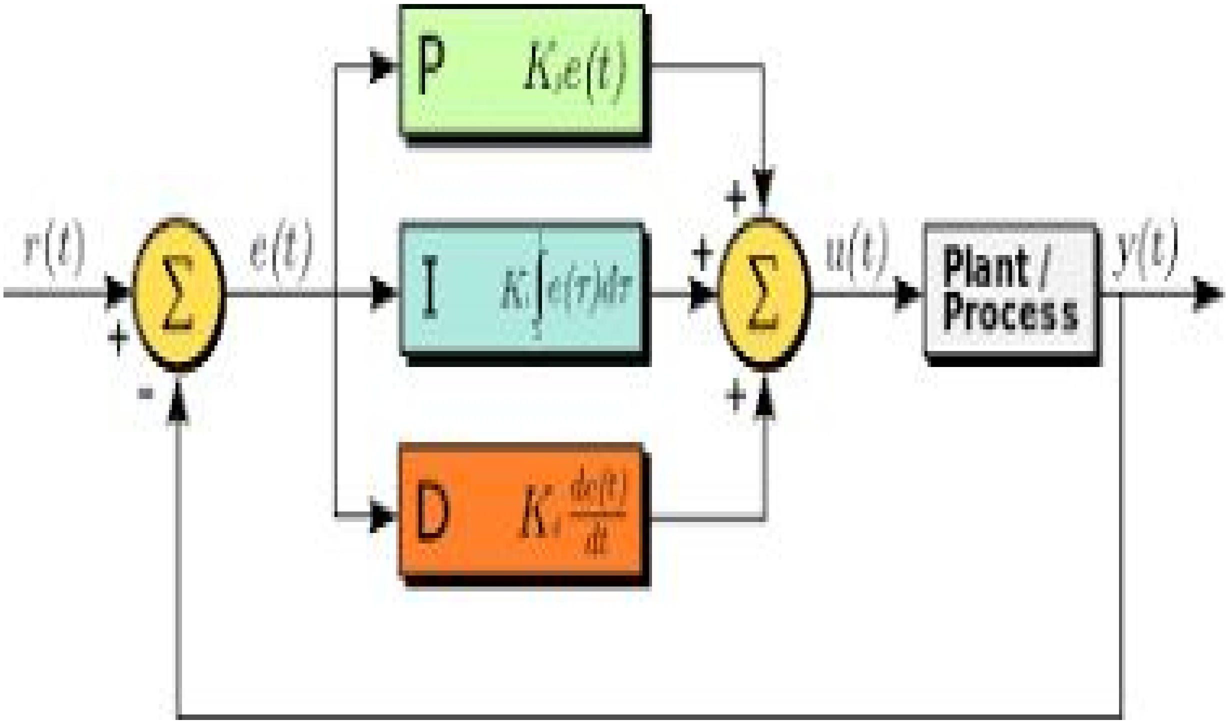
Difference between open-loop and closed loop controllers?

- The picture depicts the idea of closed loop controllers and open loop controllers.
- In **open loop controllers**, a **function maps input and output**, in **no way the output can have any say on the input**. As a user if we detect any anomalies in the output then he can intervene and change the input to get the desired output.

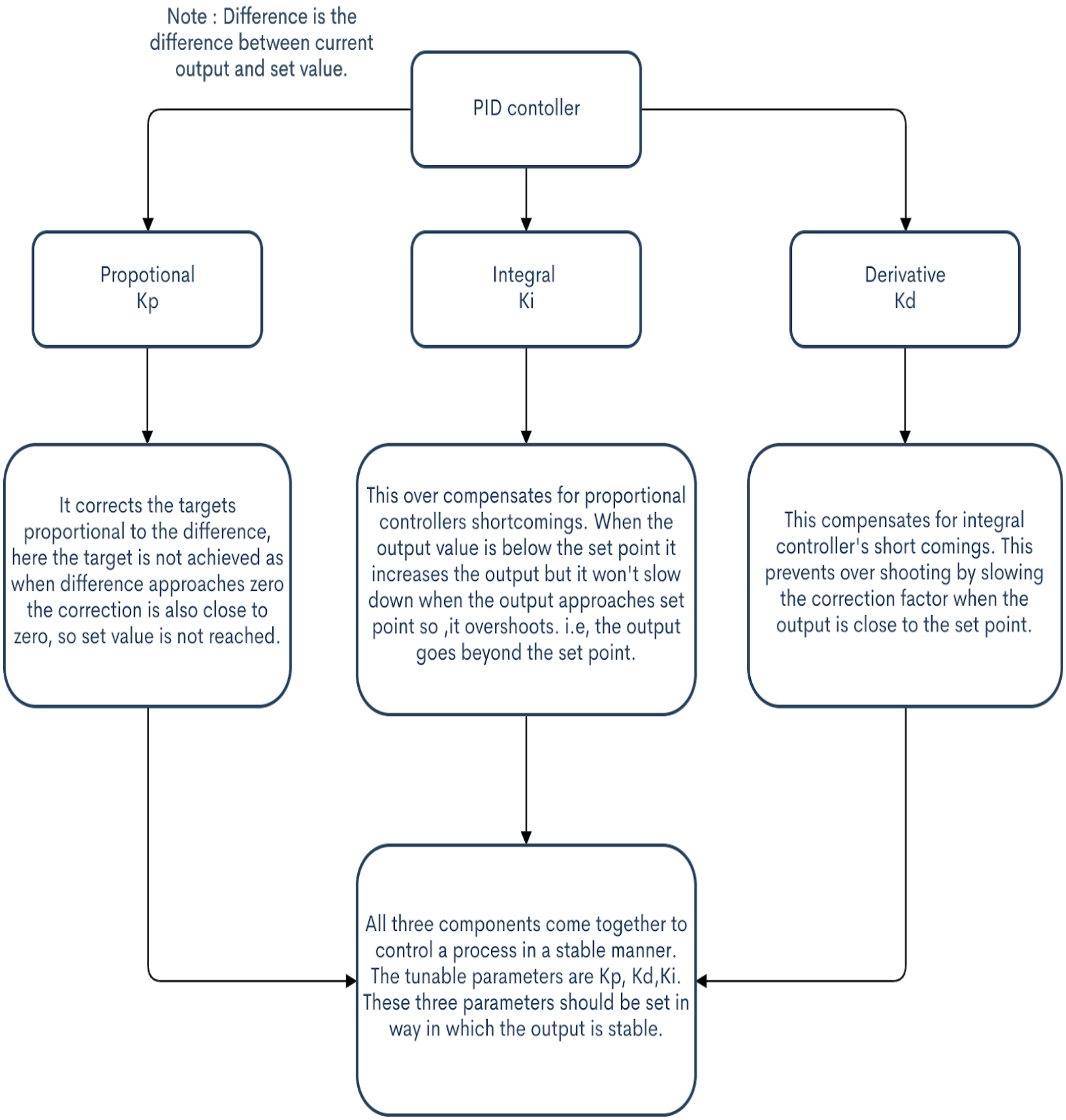


- In closed loop controllers also, **a function maps input and output but it has a parameter in the input side that can be tweaked by the change in outpu**t. A part of output is added to the input in some modified form so that the system maintains a set value without much deviation.This closed loop controller gives us control over the output. This is called feedback, it is the control action taken by the system to correct itself.This is the base for PID controllers

What are PID controllers?

- PID is basically a closed loop controller, that has a **control** loop feedback mechanism to **control** process variables.
- **The purpose of a PID controller is to force feedback to match a setpoint.**
- It is very useful in industries and applications where we need to maintain a set point value in a process.
- 

Flow chart to explain the working of pid controller.e



Note : Difference is the difference between current output and set value.

It corrects the targets proportional to the difference, here the target is not achieved as when difference approaches zero the correction is also close to zero, so set value is not reached.

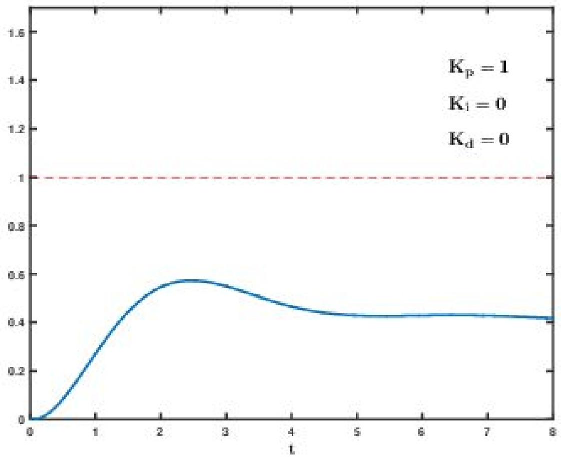
This over compensates for proportional controllers shortcomings. When the output value is below the set point it increases the output but it won't slow down when the output approaches set point so ,it overshoots. i.e, the output goes beyond the set point.

This compensates for integral controller's short comings. This prevents over shooting by slowing the correction factor when the output is close to the set point.

All three components come together to control a process in a stable manner. The tunable parameters are Kp, Kd,Ki. These three parameters should be set in way in which the output is stable.

Working of a PID algorithm

Optimum way to tune a PID controller

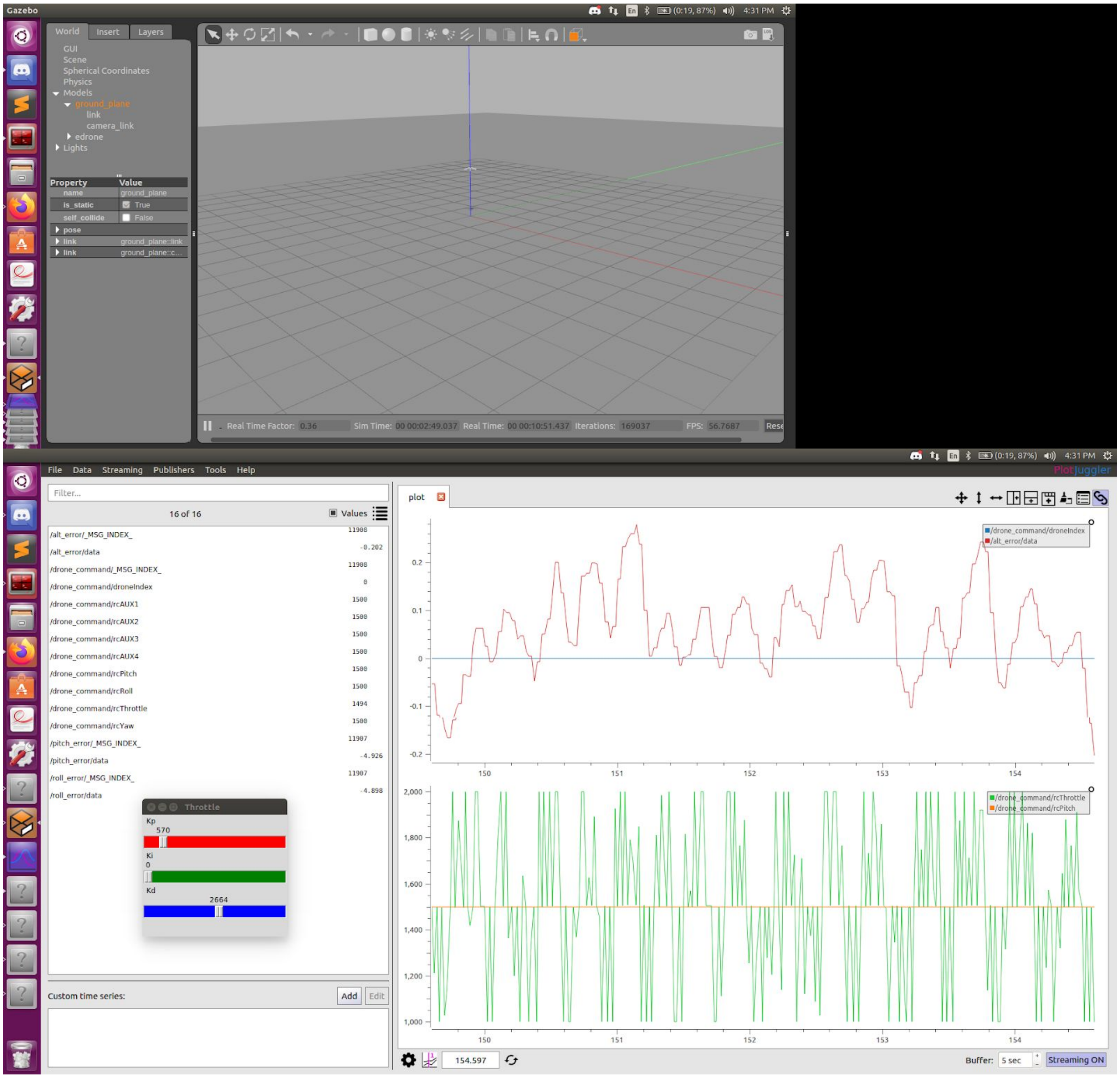


- Link for gif : https://drive.google.com/file/d/1jU-Wo4wZHKE2Oj6wej7Ept0DLSII8UaL/view?usp=sharing
- The above gif shows how changing parameters of a PID changes the stability of the output, this process of changing the parameters is called tuning.
- There are numerous ways in which a PID can be tuned, some PIDs have auto tuning where the PID tunes itself to the perfect value. New PIDs use fuzzy logic to tune the PID.
- There are classes of tuning techniques called intelligent tuning where AI is leveraged to tune the parameters of PID.
- The optimal way is subjective, there is also manual tuning, the choice of tuning comes down to the skill of the user and the requirements in the application.

Task 2

- In this task we were asked to design a PID from scratch,to balance a drone in the given set point. We had to tune kp,pd,ki for all roll, pitch and throttle.
- The videos in the below reference helps you to understand what is roll, pitch, yaw, throttle. Understanding of these parameters is very important when it comes to tuning the pid.
- We had to work as a team to come up with better solution, and we had to do trail and error which took us a considerable amount of time.
- We used plotjuggler to tune the PID, it was pretty intuitive and helped us visualze what was happening under the hood.

Plotjuggler



Property	Value
name	ground_plane
is_static	☑ True
self_collide	□ False
pose	
link	ground_plane::link
link	ground_plane::c...

Real Time Factor: 0.36 Sim Time: 00 00:02:49.037 Real Time: 00 00:10:51.437 Iterations: 169037 FPS: 56.7687

16 of 16 ■ Values

/alt_error/_MSG_INDEX_	11908
/alt_error/data	-0.202
/drone_command/_MSG_INDEX_	11908
/drone_command/droneIndex	0
/drone_command/rcAUX1	1500
/drone_command/rcAUX2	1500
/drone_command/rcAUX3	1500
/drone_command/rcAUX4	1500
/drone_command/rcPitch	1500
/drone_command/rcRoll	1500
/drone_command/rcThrottle	1494
/drone_command/rcYaw	1500
/pitch_error/_MSG_INDEX_	11907
/pitch_error/data	-4.926
/roll_error/_MSG_INDEX_	11907
/roll_error/data	-4.898

Throttle

Kp
570

Ki
0

Kd
2664

Custom time series:

154.597

Buffer: 5 sec Streaming ON

The above picture shows the output of plotjuggler, you can see that it helps in visualising the errors in real time.
Link for final output of task_2 :
https://drive.google.com/file/d/1LKb5b9EmBgk3RpLa_yY6M2xfU6uF_nEr/view?usp=sharing

References (Videos/websites/books/papers)

-https://raw.githubusercontent.com/wiki/gestom/CosPhi/papers/2015_irososar_whycon.pdf (for whycon)

-https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html (Aruco markers)

-https://arxiv.org/pdf/1709.04981.pdf#:~:text=Whycon%20is%20a%20good%20alternative,and%20for%20Whycon%2013.181%20m. (Aruco vs Whycon)

-https://circuitglobe.com/difference-between-open-loop-and-closed-loop-system.html#:~:text=One%20of%20the%20significant%20difference,control%20action%20of%20the%20system.(Open loop vs closed loop)

-https://www.omega.co.uk/prodinfo/pid-controllers.html#:~:text=A%20PID%20controller%20is%20an,most%20accurate%20and%20stable%20controller. (PID)

-https://www.omega.co.uk/prodinfo/how-does-a-pid-controller-work.html (PID)

-https://www.omega.co.uk/prodinfo/pid-controller-types.html(PID)

-https://www.omega.co.uk/prodinfo/how-to-tune-a-pid-controller.html(PID)

-https://www.semanticscholar.org/paper/Tuning-PID-Controllers-Using-Artificial-Techniques-Salem-Hassan/9167d3f20eb15c7fae66e62461070299ca2c86fd?p2df (PID tuning using AI)

Task 2 - references

- https://www.youtube.com/watch?v=R_ekdXcDQHw (Roll, pitch yaw, throttle)
- https://www.youtube.com/watch?v=bJFHZpgh6VU (Roll, pitch yaw)
- https://www.youtube.com/watch?v=AsZq5eqgtQc (ROS-error)
- https://www.youtube.com/watch?v=wkfEZmsQqiA&list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y (PID- play list)
- http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/ (PID-code-This helps us understand how PID is under the Hood)
- http://wiki.ros.org/eyantra_drone (Drone that we used in our stimulation)
- https://vimeo.com/316907214(plotjuggler)
- https://www.plotjuggler.io/cheatsheet (plotjuggler)