

What are whycon markers and how do they work?

WhyCon is a version of a vision-based localization system that can be used with low-cost web cameras, and achieves millimeter precision with very high performance. The system is capable of efficient real-time detection and precise position estimation of several circular markers in a video stream. WhyCon is meant as an alternative to widely used and expensive localization systems. It is fully open-source.

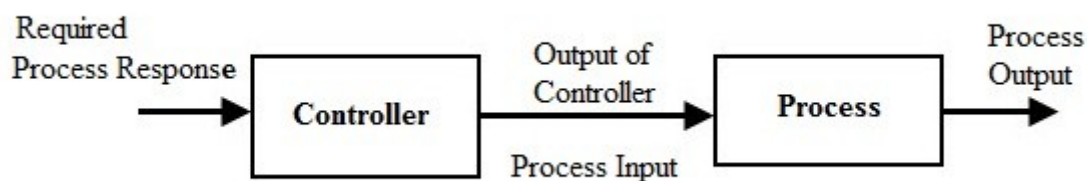
At stage 1 the algorithm perform flood fill technique and search for continuous black segments and then it The ratio of each segment's pixels to its bounding box area is then compared to a value calculated from dimensions of the whycon marker. After that it searches for the white pixels to get the centroid and calculates the 3D position .then the other whycon markers are detected and the distance between each markers are calculated and using those data the localisation of the robot is done.

What are Aruco markers and how do they work?

Aruco markers are markers which are placed on objects and scenes for uniquely identifying based on the pattern on the marker. Each marker represents different objects which makes robot easier to identify the particular object associated with the pattern

Open Loop Control System

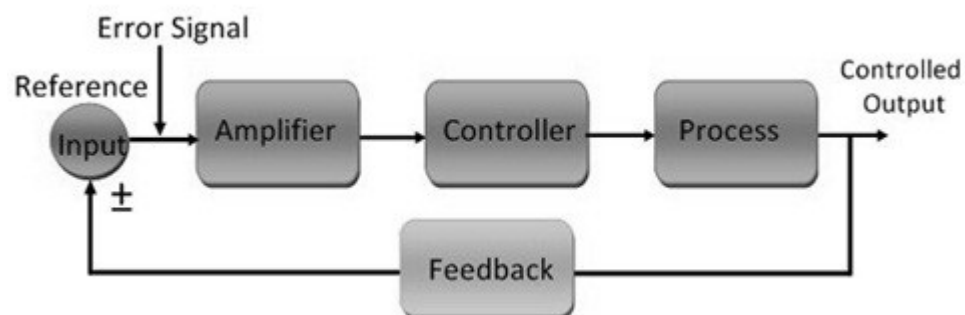
It is system in which the output of the system does not affect the action of the control system .Basically , the system that totally depends on the input and provides output based on the input .So it does not require a feedback .Since it does not check the output it is prone to have errors at the output. And the advantage of the open loop system is that it is quick to react as the control actions are based on the input



Closed-Loop Control System

In this system the action of the control systems is based on the error, which is the difference in input and output signal. As the control system requires output signal to calculate the error, this system requires a feedback node. This control system is not as fast as open loop control system as the output is based

on previous cycle(refers that the true output is produced at second cycle) .The advantage is that the errors caused due to some external factors can be reduced.



PID controllers

PID (proportional integral derivative) controllers acts based on the errors between set point and measured value. Each part in PID controller contributes for minimising the error. Each part of the PID controller has constants or multiplier and it has to be tuned properly to get desired output.

The proportional part just multiplies the error with the multiplier.

The integral part integrates over the error.

The derivative part tries to reduce the slope in order to reduce overshoot.

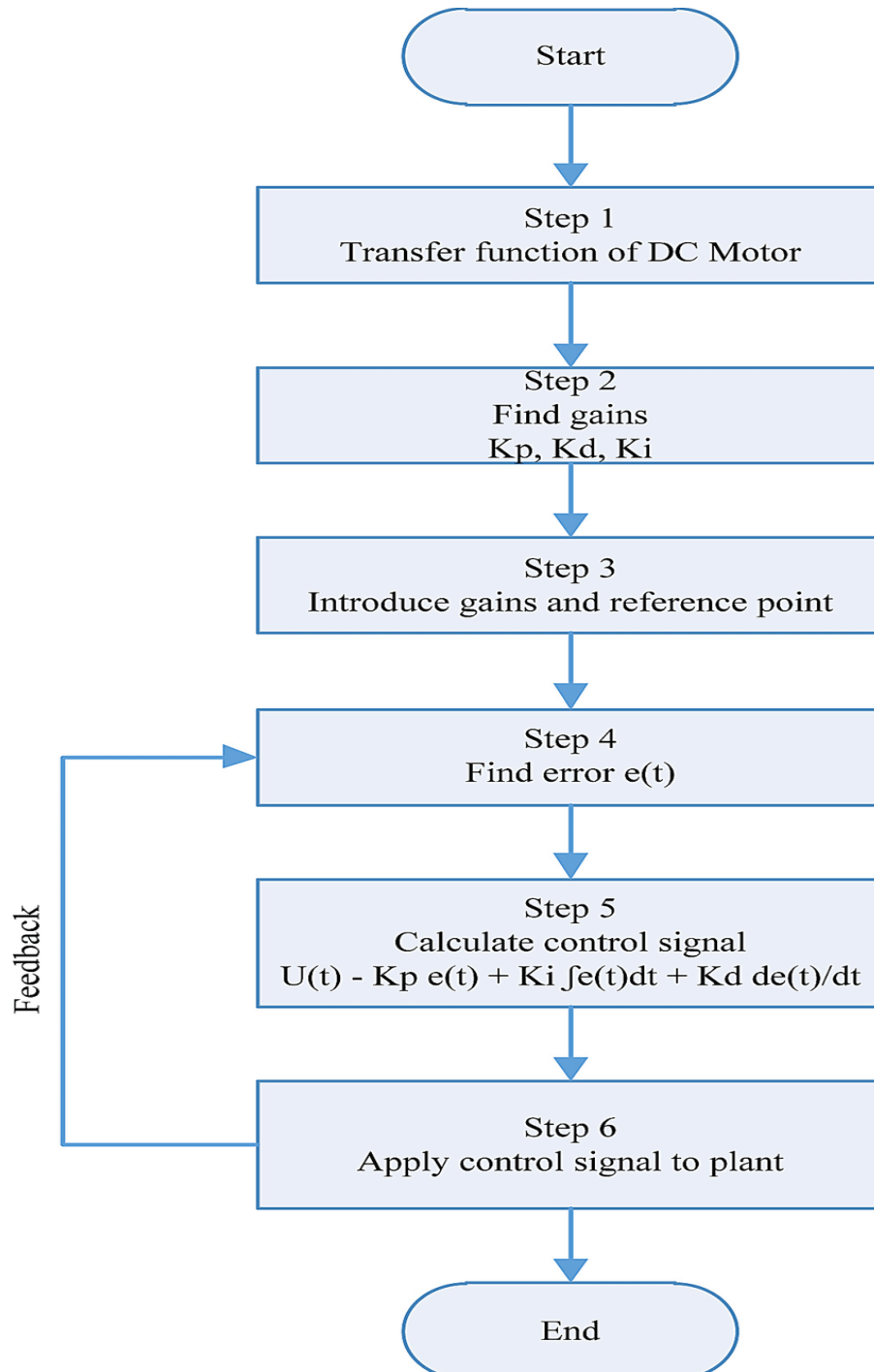
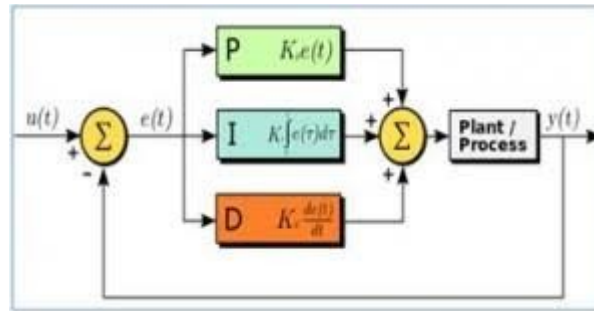
Optimal way of tuning PID controller

Rule Name	Tuning Parameters
Classic Ziegler-Nichols	$K_p = 0.6 K_u$ $T_i = 0.5 T_u$ $T_d = 0.125 T_u$
Pessen Integral Rule	$K_p = 0.7 K_u$ $T_i = 0.4 T_u$ $T_d = 0.15 T_u$
Some Overshoot	$K_p = 0.33 K_u$ $T_i = 0.5 T_u$ $T_d = 0.33 T_u$
No Overshoot	$K_p = 0.2 K_u$ $T_i = 0.5 T_u$ $T_d = 0.33 T_u$

the period T_u of the oscillation frequency at the stability limit

the gain margin K_u for loop stability

Flowchart for working of PID controller



References

Open and Closed loop control system -<https://www.elprocus.com/difference-between-open-loop-closed-loop-control-system/#:~:text=The%20main%20difference%20between%20open,depends%20on%20the%20controlled%20act.>

PID controller - https://en.wikipedia.org/wiki/PID_controller

Optimal way of tuning PID controller -
<http://www.mstarlabs.com/control/znrule.html#Ref4>

Whycon marker - <https://core.ac.uk/download/pdf/42583963.pdf>

AruCo markers - <https://www.learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/#:~:text=An%20aruco%20marker%20is%20a,in%20a%20variety%20of%20sizes.>

PID flowchart1 -
https://upload.wikimedia.org/wikipedia/commons/thumb/4/43/PID_en.svg/400px-PID_en.svg.png

PID flowchart2 -
https://www.researchgate.net/profile/Sohaib_Aslam2/publication/307092705/figure/fig3/AS:400324838871042@1472456287510/Flow-Chart-of-PID-Implementation.png