Some basics before we start

These are some terminologies and information that should be considered before getting in.

WhyCon

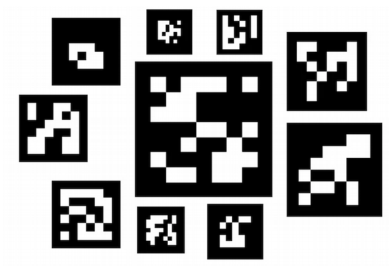
A precise, efficient and low-cost localization system

WhyCon is a version of a vision-based localization system that can be used with low-cost web cameras, and achieves millimiter 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.

The proposed system is based on a fast and precisedetection of a black and white roundel, which consists of twoconcentric annuli with a white central disc. An initial detectionstep is performed to identify the position of the roundel inpixel coordinates. Using camera re-projection techniques andknown dimensions of the inner and outer roundels, the three-dimensional position of the target with respect to the camera iscomputed. Finally, a transformation of the coordinate frame isapplied to compute target coordinates with respect to a user-defined frame, either in three or two dimensions, dependingon the chosen scenario.

What are ArUco markers?

ArUco markers were originally developed in 2014 by S.Garrido-Jurado et al., in their work "Automaric generation and detection of highly reliable fuducial markers under occlusion". ArUco stands for 'Augmented Reality University of Cordoba'. That is where it was developed in Spain. Below are some examples of the ArUco markers.



An aruco marker is a *fiducial marker* that is placed on the object or scene being imaged. It is a binary square with black background and boundaries and a white generated pattern within it that uniquely identifies it. The black boundary helps making their detection easier. They can be generated in a variety of sizes. The size is chosen based on the object size and the scene, for a successful detection. If very small markers are not being detected, just increasing their size can make their detection easier.

Difference between Open-Loop and Closed-Loop control System

S.No.	Open-Loop Control System	Closed-Loop Control System
1.	It easier to build.	It is difficult to build.
2.	It can perform better if the caliberation is properly done.	It can perform better because of the feedback.
3.	It is more stable.	It is comparatively less stable.
4.	Optimization for desired output can not be performed.	Optimization can be done very easiy.
5.	It does not consists of feedback mechanism.	Feedback mechanism is present.
6.	It requires less maintenance.	Maintenance is difficult.

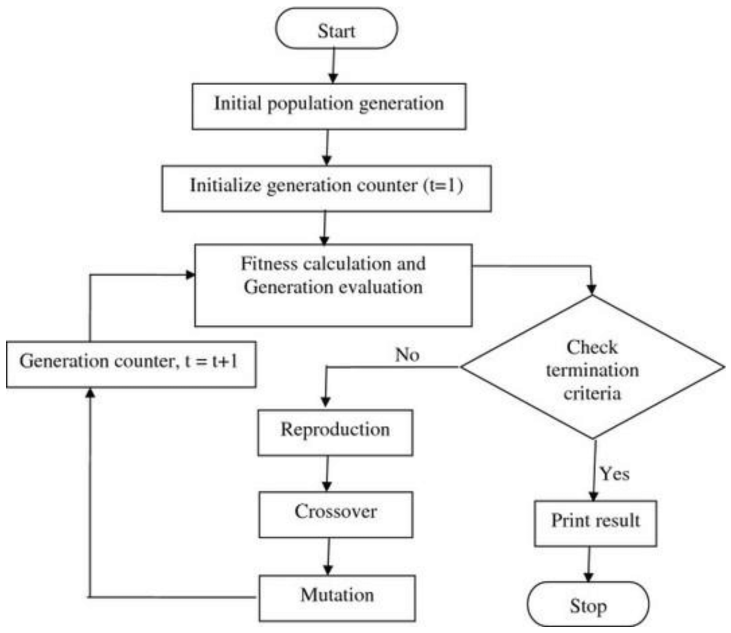
S.No.	Open-Loop Control System	Closed-Loop Control System
7.	It is less reliable.	It is more reliable.
8.	It is comparatively slower.	It is faster.
9.	It can be easily installed and is economical.	Complicated installation is required and is expensive.

PID Controller :-

A **proportional–integral–derivative controller** (**PID controller** or **three-term controller**) is a control loop mechanism employing feedback that is widely used in industrial control system and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an *error value* as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on propotiona,l, integral, and derivative terms (denoted *P*, *I*, and *D* respectively), hence the name.

In practical terms it automatically applies accurate and responsive correction to a control function. An everyday example is the cruise control on a car, where ascending a hill would lower speed if only constant engine power were applied. The controller's PID algorithm restores the measured speed to the desired speed with minimal delay and overshoot by increasing the power output of the engine.

Flow-chart to explain the working of PID Controller


Start
Initial population generation
Initialize generation counter (t=1)
Fitness calculation and Generation evaluation
Check termination criteria
No
Yes
Generation counter, t = t+1
Reproduction
Crossover
Mutation
Print result
Stop

Setting PID tuning parameters

Every process has unique characteristics, even when the equipment is essentially identical. The PID settings must be selected to suit these local differences.

In broad terms, there are three approaches to determine the optimal combination of these settings: manual tuning, tuning heuristics, and automated methods.

Manual tuning of pid controller

Manual PID tuning is done by setting the reset time to its maximum value and the rate to zero and increasing the gain until the loop oscillates at a constant amplitude. (When the response to an error correction occurs quickly a larger gain can be used. If response is slow a relatively small gain is desirable). Then set the gain of the PID controller to half of that value and adjust the reset time so it corrects for any offset within an acceptable period. Finally, increase the rate of the PID loop until overshoot is minimized.

Tuning Heuristics

Many rules have evolved over the years to address the question of how to tune a PID loop. Probably the first, and certainly the best known are the Zeigler-Nichols (ZN) rules.

First published in 1942, Zeigler and Nichols described two methods of tuning a PID controller. These work by applying a step change to the system and observing the resulting response. The first method entails measuring the lag or delay in response and then the time is taken to reach the new output value. The second depends on establishing the period of a steady-state oscillation. In both methods, these values are then entered into a table to derive the values for gain, reset time and rate for the control system.

ZN is not without issues. In some applications it produces a response considered too aggressive in terms of overshoot and oscillation. Another drawback is that it can be time-consuming in processes that react only slowly. For these reasons, some control practitioners prefer other rules such as Tyreus-Luyben or Rivera, Morari and Skogestad.

Auto Tune

Most process controllers sold today incorporate auto-tuning functions. Operating details vary between manufacturers but all follow rules similar to those described above. Essentially, the PID controller "learns" how the process responds to a disturbance or change in set point, and calculates appropriate PID settings. By observing both the delay and rate with which the change is made it calculates optimal P, I and D settings, which can then be fine-tuned manually if needed.

Newer and more sophisticated PID controllers, incorporate fuzzy logics with their auto tuning capabilities. This provides a way of dealing with imprecision and nonlinearity in complex control situations, such as are often encountered in manufacturing and process industries, and helps with tuning optimization.

Refferences

https://github.com/lrse/whycon/blob/master/README.md#references

https://www.learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/

https://www.wikipedia.org/

https://www.geeksforgeeks.org