

# Controlling the Position and Velocity in Space of the Quad-Rotor UAV AR.Drone Using Predictive Functional Control and Image Processing in Open CV

Anusha Gururaj Jamkhandi <sup>1+</sup>, Saeed Tulpule <sup>1</sup>, Ashvini Chaturvedi <sup>1</sup> and Jean-Noel Charvet <sup>2</sup>

<sup>1</sup> National Institute of Technology, Surathkal, Karnataka, India

<sup>2</sup> Ecole Catholique d'Arts et Metiers, Lyon, France

**Abstract.** The aim of this work is to use Predictive Functional Control (PFC) to make the Parrot Drone follow a red ball. The Drone has an in-built proportional-integral-derivative (PID) controller in its software development kit to control its motion. However, large time delays and high-order dynamics are the common dynamic characteristics that are difficult to control using PID technique. This work aims to use PFC instead of PID control due to its shortcomings. We begin with developing an equivalent first order process model for Drone motors and its PID controllers by analyzing the Euler angles (pitch, roll and yaw) and the velocities in the X and Y directions. The Drone is then made to follow a red ball for which the coordinates of its centre are determined through image processing using Hough circles in Open CV and subsequently these coordinates are passed to the PF controller. The X and Y positions are controlled by the PFC by manipulating pitch and roll angles respectively. The results obtained are quite promising and thereby an attempt is made to overcome the limitations of PID controllers to a significant degree by using PF controllers.

**Keywords:** proportional-integral-derivative control, predictive functional control, euler angles, hough circles, set point, manipulated variable, controlled variable.

## 1. Introduction

The AR. Drone (Version 1.7) is a quad-rotor UAV (unmanned aerial vehicle) that receives commands like the real time information about its current state, current position and orientation, and video stream of a selected camera via a wireless link. The mechanical structure comprises of four rotors attached to the four ends of a crossing to which the battery and the RF hardware are attached. Each pair of opposite rotors turns in the same direction [4]. One pair turns clockwise and the other anticlockwise. The AR.Drone Software Development Kit (SDK) allows third party users to remotely control the Drone from any personal computer with a Wi-Fi Connection. Controlling and configuring the Drone is done by sending AT commands on UDP port 5556 periodically (usually 30 times per second) [1,5]. Information about the drone (like its status, its position, speed, engine rotation speed, etc.), termed as navdata, is sent by the Drone to its client on UDP port 5554. PID control, which is most commonly used in today's industrial control systems, is used to control the motion of the Drone. However, PID controllers when used alone exhibit sluggish behavior while responding to large systems thus giving rise to unsatisfactory performances. Another problem faced with PID controllers is that they are linear, and in particular, symmetric. Thus, the performance of PID controllers in non-linear systems (such as HVAC systems) is variable. Hence, PF control is preferred in such systems which are accompanied by large delays and higher ambient noises. We propose to use PFC or model predictive control (MPC) instead of PID control for the motion of the Drone. PFC is an advanced method of process control that has been extensively in use in the process industries such as chemical plants and oil refineries. PFCs rely on dynamic models of the process, most often linear empirical models obtained by system

---

<sup>+</sup> Corresponding author. Tel: + (91)9663931535.  
E-mail address: [anusha.gururaj@gmail.com](mailto:anusha.gururaj@gmail.com).

identification [2]. The models used in PFC are generally intended to represent the behavior of complex dynamical systems and they predict the change in the dependent variables of the modeled system that will be caused by changes in the independent variables. In the case of the Drone, the independent variable is the pitch angle of drone which is manipulated by us, and the dependent variable is the position of the Drone that results from change in pitch.

## 2. Methodology

### 2.1. Image processing in open CV

The Drone has two cameras: the front camera (60X480 pixel color) and the bottom camera (176X144 pixel color). We use the front camera, which is actually a CMOS sensor with 90 degree angle lens for the image processing. A video stream is sent by the AR.Drone to the client device on port 5555. Images from this video stream can be decoded using the codec included in the SDK. An Open CV image is first created frame-wise (frame rate 15Hz) from the image obtained from the Drone's front camera [7]. Color based tracking of the red ball is done by filtering the red color from the image by converting it into HSV color space. The image is then smoothened and the shape of the circle is detected by shape based tracking using Hough transform [8]. The minimum and maximum HSV ranges for red are taken as (20,100,100) and (30,255,255) respectively. Since the image processing is done by accessing the front camera of the Drone, the pitch, roll and gaz commands are with respect to the vertical plane. The centre of the Hough circle so obtained is passed as the set point to the Predictive Functional Controller.



Fig. 1: Front Camera of the Drone



Fig. 2: Centre of the Hough circle

### 2.2. Modeling the drone

The Parrot Drone has a 6-degree-of-freedom inertial measurement unit comprising of a MEMS 3-axis accelerometer, 2-axis gyro and a single-axis yaw precision gyro meter and a sonar based altimeter with a range of 6 meters [6,9,10]. It has a built-in PID controller that controls the motion of its four motors [3]. While modeling the Drone into a linear system, we reduce the integral and derivative components of the PID controller to zero. This Drone model acts as the process for the PFC controller. The information about the values of the Euler angles and the velocities in X and Y directions sent by the Drone are plotted and analyzed in Matlab. The position at any instant relative to the initial position is obtained by integrating velocity values over the time period and is shown in Fig. 3.



Fig. 3: Plots of data sent by Drone

### 2.3. Finding the time constant and the delay from the curve

Fig 4 shows the graph of position of the Drone with a PID controller for a closed loop and therefore the step response (pitch) is a stable response (response to a first order) We notice that the minimum value of position is 1.7mm (at  $t=8.045\text{sec}$ ) and maximum value is 3137mm (at  $t=12.48\text{sec}$ ) resulting in a total change in position of 3318.7mm. 63% of this value evaluates to 2090.7mm. As we start the movement at  $X= -181.7\text{mm}$  the value of  $X$  at which the distance covered is 2090.7mm is 1909mm and the time corresponding to this  $X$  is 9.68sec. Hence we get the value of time constant which is the difference between 9.68 and 8.045 to be 1.63sec.

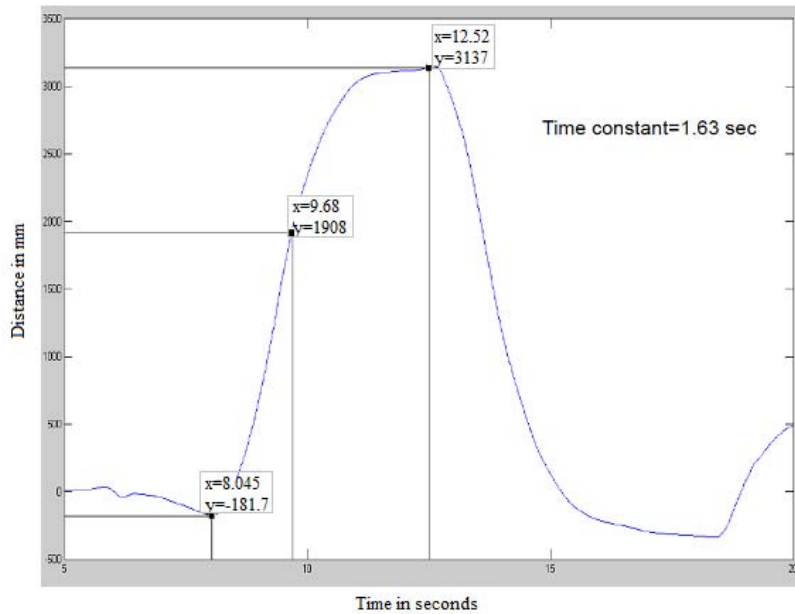


Fig. 4: Graph used for calculating the time constant

### 2.4. Designing predictive functional control for the drone

The transfer function from pitch to position is integrative in nature and hence we cannot use step PFC to control the Drone (Fig. 5). For controlling the drone with PFC, we create a thread which calls the PFC function every 100ms. A Visual C++ code is written for the timer where the  $X$  position of the Drone is initialized as the Controlled Variable of the Drone. The Manipulated Variable is given as the pitch value in the AT commands sent to control the drone. Similar approach is followed to control  $Y$  position by manipulating the roll value.

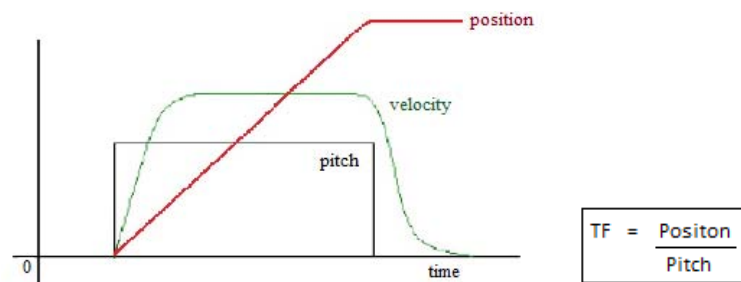


Fig. 5: Transfer function

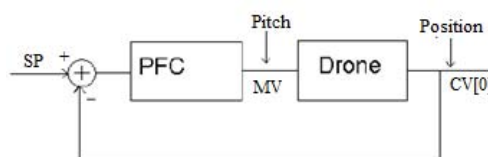


Fig. 6: PFC model for the Drone

## 2.5. Results and conclusion

Fig. 7 shows the drone output window where  $SP_x$  and  $SP_y$  represent the set points which correspond to the X and Y coordinates of the center of the ball respectively. Fig. 8 shows the results of PFC vs PID with respect to delay and noise. It can be inferred that a PF controlled process reaches its set point faster than a PID controlled one. Also, as shown in Fig. 9, in case of noisy systems, PFC gives better results. On analyzing the motion of the Drone with PFC, it was seen that this type of control made the Drone more stable and its motion more accurately predictable. Hence PFC renders to be a better control technique than PID control.

```

Parrot A.R. Drone SDK Demo for Windows
=====
Navdata for flight demonstrations
=====
Control state : CTRL_LANDED
Battery level : 85/100
Orientation : [Theta] 789.000 [Phi] 639.000 [Psi] 3319.000
Altitude : 221
X position : 19.506cm Y position : -8.729cm MV_x : 0.004 MV_y : 0.003
Speed : [vX] 157.800 [vY] 96.340 [vZ] 0.000

[Pitch 0.000000] [Roll 0.000000] [Yaw 0.000000] [Gaz 0.000000]
[SP_x 50.000] [SP_y -50.000]

```

Fig. 7: The Drone output window.

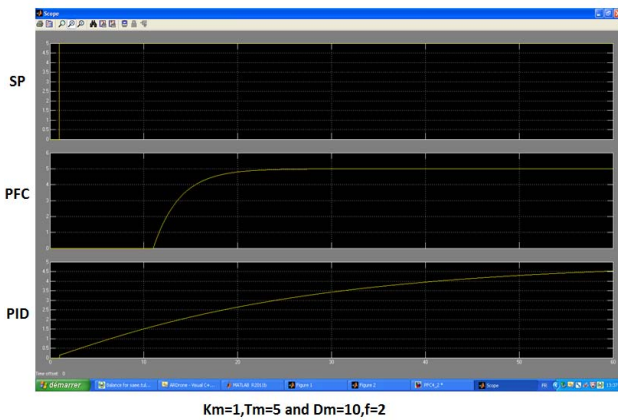


Fig. 8: PFC vs PID with delay

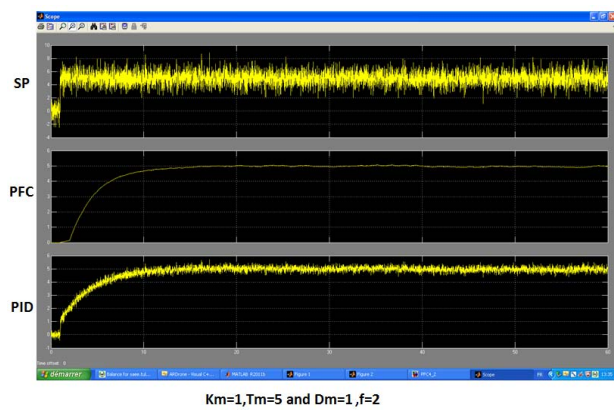


Fig. 9: PFC vs PID with noise

## 3. Acknowledgement

We thank our project mentors Prof. Jean-Noel CHARVET, Prof. Christophe JOUVE, Prof. Helen and Prof. Olivia PETILLOT for their continuous guidance throughout the project. We thank ECAM, Lyon for providing us with the environment and necessary resources for the project. We also thank Prof. Umesh of National Institute of Technology Karnataka for his support.

## 4. References

- [1] S. P. N. Brulez and P. Eline. ARDrone\_SDK\_1\_7\_Developer\_Guide, 2011.
- [2] J. Richalet, D. O. Donovan. Predictive Functional Control: Principles and Industrial Applications, Springer Verlag London Limited, 2009.
- [3] M. D. L. C. D. Oliveira. Master's thesis: Modeling, Identification and Control of a Quadrotor Aircraft, Czech Technical University, Prague, June 2011.

- [4] J. V. Portal. A java autopilot for Parrot A.R. Drone Designed with DiaSpec, Universidade Federal Do Rio Grande Do Sul.
- [5] T. Krajn'ik, V. Von'asek, D. Fi'ser, and J. Faigl. AR-Drone as a Platform for Robotic Research and Education, Czech Technical University, Prague.
- [6] N. Dijkshoorn and A. Visser. Integrating sensor and motion models to localize an autonomous AR.Drone, Universiteit van Amsterdam, *International Journal of Micro Air Vehicles*, November 1, 2011.
- [7] Gauth. Use OpenCV with the AR.Drone SDK, September 16<sup>th</sup>, 2011 (<http://gauth.fr/2011/09/use-opencv-with-the-ar-drone-sdk/>)
- [8] Utkarsh. Tracking colored objects in OpenCV, July19th, 2010 (<http://www.aishack.in/2010/07/tracking-colored-objects-in-opencv/>)
- [9] *InvenSense IDG-500 Dual-Axis Gyro*. Datasheet
- [10] *Microchip MCP6L01/1R/1U/2/4*. Datasheet



**Anusha Gururaj Jamkhandi** is currently pursuing 7<sup>th</sup> semester B.Tech in Electrical and Electronics Engineering at National Institute of Technology Karnataka, India. Her research interests include signal processing and microcontrollers. She worked extensively on EFM32 microcontroller as an intern at the Indian Institute of Science, Bangalore, India in 2011. She has successfully completed several projects in signal processing such as building an automatic speaker recognition system, building a ball follower robot, implementing Change Vector Analysis algorithm for change detection in Geographic Information Systems in Matlab, etc. This project on the AR. Drone was carried out at ECAM, Lyon in 2012 where she worked as summer exchange students. She is currently working on developing a GUI for solving different electrical and mechanical systems using different numerical integration techniques and analyzing the stability and stiffness aspects of the same.