Target Tracking by a Quadrotor Using a Combination of a Noise-Based Filter and Sensor Fusion Using a Sigmoid Function ⋆

Varun Nayak * **Raksha Rao Karaya** **

* *Birla Institute of Technology and Science, Pilani - K K Birla Goa Campus (e-mail: f20140086@goa.bits-pilani.ac.in).*
** *Indian Institute of Technology Madras (e-mail: raksharaok18@gmail.com).*

Abstract: The use of micro-class Unmanned Aerial Vehicles (UAV) for consumer applications is rapidly growing. Many such applications employ intelligent systems in order to interact with the environment around the UAV. This paper demonstrates the modelling, simulation and implementation of a one dimensional object tracking quadrotor that can detect and follow a solid object in front of it by maintaining a fixed distance in between. A novel combination of a noise-based filter along with a sensor fusion technique using a sigmoid function was developed for a specific combination of two proximity sensors. The system uses a Proportional-Integral-Derivative (PID) controller to generate a single high-level pitch command based on the filter output to track a target. Low-level attitude control and altitude maintenance is simultaneously performed by the commercially available APM 2.5 autopilot.

Keywords: Target Tracking, Sensor Fusion, Filtering Techniques, PID Control, Autonomous Mobile Robots, Proximity Sensing

1. INTRODUCTION

UAVs have always been pertinent in the domain of defense and military applications. However, nowadays, there has been an increasing trend in the use of micro-class UAVs for consumer applications such as package delivery, aerial videography, aerial inspection and so on. Owing to its simple mechanical structure, the *quadrotor platform* (four propellers) is the most common aerial vehicle configuration that is being used today. Research and development in the guidance, navigation and control of quadrotors has gained impetus due to the large scale availability of sensors, actuators and microcontrollers in recent times. Through this, UAVs are being given capabilities to intelligently perceive and act on its surrounding environment.

The primary problem of controlling a quadrotor involves state estimation and motor control in order to maintain desired attitude and thrust. Many researchers have solved this problem employing different control techniques, from classical PID or LQR control(Li and Li (2011), Salih et al. (2010), Argentim et al. (2013)) to more complex non-linear integral backstepping control (Tian et al. (2016), Bouabdallah and Siegwart (2007), Fang and Gao (2011)). More complicated approaches are taken to improve robustness by including gyroscopic effects, aerodynamic effects, etc. However, this problem is sufficiently solved and has enabled researchers to develop systems which pertain to higher level tasks like object recognition and subsequent tracking or collision avoidance.

⋆ This work was sponsored by the *Department of Aerospace Engineering, Indian Institute of Technology Madras* as a part of their summer research fellowship programme for undergraduate students.

The problem of detection, avoidance and tracking is fundamental to autonomy. Boudjit and Larbes (2015) and Dang et al. (2013) have developed object detection and target tracking system using computer vision on the commercially available AR.Drone. Mth et al. (2016) have used a similar visual servoing technique with an Extended Kalman Filter (EKF). Teulire et al. (2011) and Jurado et al. (2012) have both developed novel color-based robust target tracking techniques using visual feedback. However, all of the above works depend on a wireless link to transmit video feed and ground station processing. Blösch et al. (2010) developed a SLAM algorithm which also requires external processing.

There also have been some efforts to develop fully on-board systems. Scherer et al. (2008) used a learning approach to avoid obstacles detected using a laser scanner. Shen et al. (2011) fused data from a laser scanner and a camera with the iterative closest point (ICP) method and EKF for position estimation, to run completely on-board and uses a 1.6 GHz Atom board. Similar solutions from Engel (Engel et al. (2012a),Engel et al. (2012b)) and Celik et al. (2009) depend on an external laptop. Becker et al. (2012) used four ultrasonic sensors and a vertical camera which is effective but does not cover 360° nor perform distance control. Kendall et al. (2014) have developed a low cost on-board monocular vision based object tracking drone but fails when there is insufficient lighting. Finally, Gageik et al. (2015) uses a novel combination of an ultrasonic and an infrared sensor which is similar to the approach presented here, but was developed and tuned for collision avoidance and not object tracking.

Preprint submitted to Third IFAC International Conference on Advances in Control and Optimization of Dynamical Systems. Received September 30, 2017.

In this paper, the mathematical model of a quadrotor tracking an object in front of it in one dimension is developed, along with a PID control strategy for it. The proximity sensing is performed using two ultrasonic sensors with different operating characteristics. The reliability of the proximity data is vastly improved by a combination of noise-based filter (NBF) and an experimentally derived sensor fusion technique based on sigmoid functions. This computationally inexpensive method was developed and test on the low-cost microcontroller Arduino® UNO [1] and the commercially available APM 2.5 autopilot [2], entirely on-board the quadrotor. Since it depends only on ultrasound signals, this system becomes effective in poorly lit environments as well.

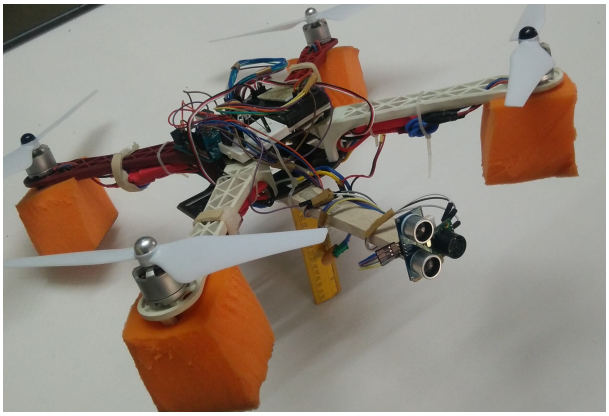

Fig. 1. Fully Assembled Quadrotor with Two Front Mounted Proximity Sensors

2. QUADROTOR PLATFORM AND SENSOR SUITE

The Quadrotor platform used in the development of this system is the low-cost, commercially available DJI F-450 kit. It is lightweight and has ample space for onboard electronics.

COMPONENT	SPECIFICATIONS
Frame Type	DJI 'X' configuration 450 mm diagonal
Motor Type	Three Phase Brushless DC, 1200 KV
ESC	30 A Rating with BEC
Battery	3 S Lithium Polymer 3500 mAh
Transmitter (TX)	6 Channel Fly-Sky FS-T6
Receiver (RX)	6 Channel Fly-Sky PWM FS R6B

Table 1. Specifications of the F-450 Quadcopter kit

In addition to the kit, the quadrotor is equipped with APM 2.5 for low-level control as well as the Arduino UNO as a companion board to send high level control commands using sensor data. The two proximity sensors were mounted on the front of the quadrotor facing forwards as seen in Fig. 1.

3. SYSTEM DYNAMICS

The quadrotor utilizes APM's altitude hold funtion to constrain itself to a single height from the ground and enables a one dimensional model. Hence, the quadrotor as show in Fig. 2 is assumed to be static in the vertical direction. The analysis that follows is a simplified model in order to estimate controller parameters and verification of controller design.

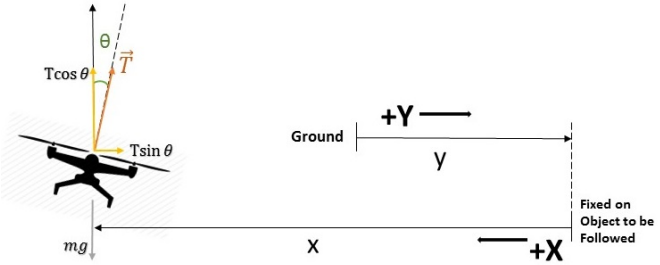

Fig. 2. System Dynamics of the Quadrotor Tracking an Object

One coordinate frame is attached to the ground and measures the distance of the object y and another frame attached to the object measures the distance of the quad x from it. Applying Newton's 2nd law on the quadrotor in the vertical and horizontal directions respectively, we get

$$mg = T\cos\theta(t) \tag{1}$$

$$m\frac{d^2x}{dt^2} + T\sin\theta(t) = m\frac{d^2y}{dt^2} \tag{2}$$

where $\theta(t)$ is the pitch angle, T is the value of the thrust, m is the total mass of the quadrotor and g is the acceleration due to gravity. Assuming that the small angle approximation is applicable without much error (since we are restricting the control input to a maximum of 12° pitch angle), we can develop the dynamic equation further into a LTI system.

$$\frac{d^2x}{dt^2} + g\theta(t) = \frac{d^2y}{dt^2} \tag{3}$$

Taking the Laplace Transform and replacing $\theta(t)$ as a linear function of the input PWM signal [3] we get

$$\frac{X(s) - Y(s)}{P(s)} = \frac{gk}{s^2} \tag{4}$$

where $P(s)$ is the Laplace transform of the PWM control input signal and k is the constant of proportionality in the linear relationship between the PWM signal and the pitch angle given by $\theta(s) = kP(s)$

4. OBJECT DETECTION AND DISTANCE ESTIMATION

4.1 Proximity Sensor Specifications and Comparison

The two ultrasonic proximity sensors used are the HC-SR04 and the Maxbotix® MB1220. Both are cheap and easy to integrate with the Arduino environment.

[1] https://www.arduino.cc/
[2] http://ardupilot.org/copter/
[3] determined using Mission Planner client software for APM and http://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html

Preprint submitted to Third IFAC International Conference on Advances in Control and Optimization of Dynamical Systems. Received September 30, 2017.

SPECIFICATION	HC-SR04	Maxbotix® MB1220
Range	3 cm-400 cm	20 cm-760 cm
Resolution	3 mm	1 cm
Measuring Angle	Low, <15°	High, <30°
Reliable Range for small objects	3 cm-40 cm	20 cm-150 cm
Noise filtering	Noisy with sharp peaks	Good over a large range above 25 cm

Table 2. Comparison between the HC-SR04 and the Maxbotix MB1220 sensors based on datasheet values and manual testing

From the comparison given in Table 2, it is evident that the two sensors have many distinct operational characteristics. Therefore, a decision to employ an effective sensor fusion method along with data filtering was taken.

4.2 Data Filtering

Since a stable, reliable and accurate stream of distance data is required as input to the controller, the sensor data was processed through a feasible filter. A comparative analysis of three filters - exponential smoothing, digital filter and the noise-based filter was done by implementation on hardware in order to choose the best possible candidate for the final system.

Exponential Smoothing: Exponential Smoothing is a simple low-pass filter. Let $s[n]$ be the raw sensor data obtained at the nth time sample and $p[n]$ be the filtered data calculated using the nth time sample.

$$\begin{aligned} p[0] &= s[0] \\ p[1] &= \alpha s[1] + (1-\alpha)p[0] \\ p[n] &= \alpha s[n] + (1-\alpha)p[n-1] \end{aligned} \tag{5}$$

where α is the *smoothing factor*.

Digital Filter: The digital low-pass filter uses a time constant τ and the loop time T to calculate the filtered output. Let $y[n]$ be the filtered output and $x[n]$ be the raw sensor data.

$$y[n] = ay[n-1] + (1-a)x[n] \tag{6}$$

where $a = \frac{\tau}{\tau+T}$. The constant τ depends on the desired cut-off frequency for the filter. The loop time was manipulated by taking readings at specific intervals of time, depending on the update frequency of the sensor in use.

Noise-Based Filter (NBF): The NBF designed specifically for this system recursively calculates the estimate of distance combining prior predictions, the sensor model and noisy measurements. The scalar variable distance is measured by the sensor and is estimated using an estimate variable which is equal to the best estimate or the filtered data i.e. the output of the system. It is essentially a low-pass filter with a varying time constant based on a Gaussian model. We define a gain $G[n]$ to distribute weightage between the previous estimate and the current sensor reading. At the n^{th} time step, the gain is updated using the sensor variance σ_s^2 (which is known from intrinsic sensor characteristics). $\sigma_x^2[n]$ is the estimate variance at the n^{th} time step.

$$G[n] = \frac{\sigma_x^2[n]}{\sigma_s^2 + \sigma_x^2[n]} \tag{7}$$

Next, the new estimate $x[n]$ is calculated using the previous estimate $x[n-1]$ (starting from an arbitrary initial condition that corrects itself) and the current raw sensor reading.

$$x[n] = (1 - G[n])x[n-1] + G[n]S[n] \tag{8}$$

where $S[n]$ is the raw sensor data. Next, the estimate variance is updated using the previous variance and the gain as shown below.

$$\sigma_x^2[n] = (1 - G[n])\sigma_x^2[n-1] \tag{9}$$

Comparison and Modification: For comparison between the performance of the sensors, each algorithm was implemented on hardware and the data was plotted on MATLAB® for observation and qualitative judgement.

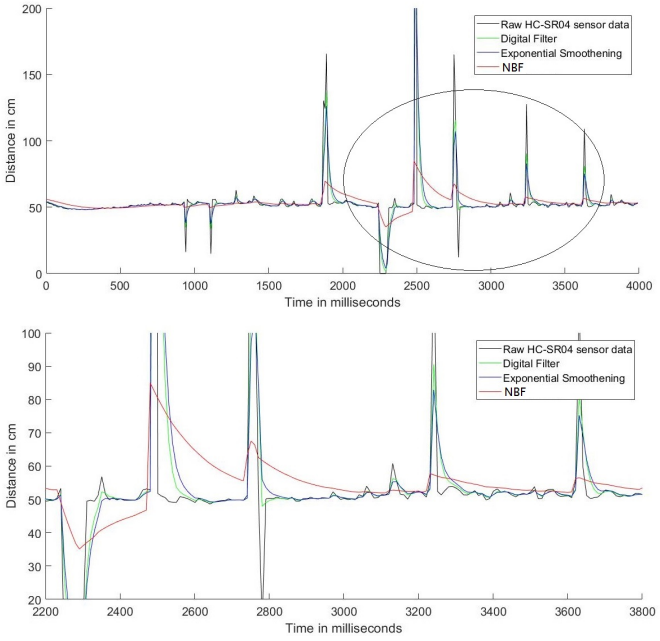

Fig. 3. Comparison between raw data and various filtered data for the HC-SR04 sensor with blown-up view on the bottom. The bottom figure clearly shows capability of the NBF to substantially reduce deviation from the 'true' value of proximity.

From Fig. 3 and Fig. 4, it can be seen that the output from the exponential smoothing and the digital filter contain noise of large magnitude. The NBF rejects this noise very effectively in comparison to the other two filters. However, it is sluggish because of a relatively low sensor update frequency. Therefore, the NBF gain was modified with an additive factor 'A' while ensuring that it stayed within the lower limit of 0 and the upper limit of 1.

$$G[n] = \frac{\sigma_x^2[n]}{\sigma_s^2 + \sigma_e^2[n]} + A \tag{10}$$

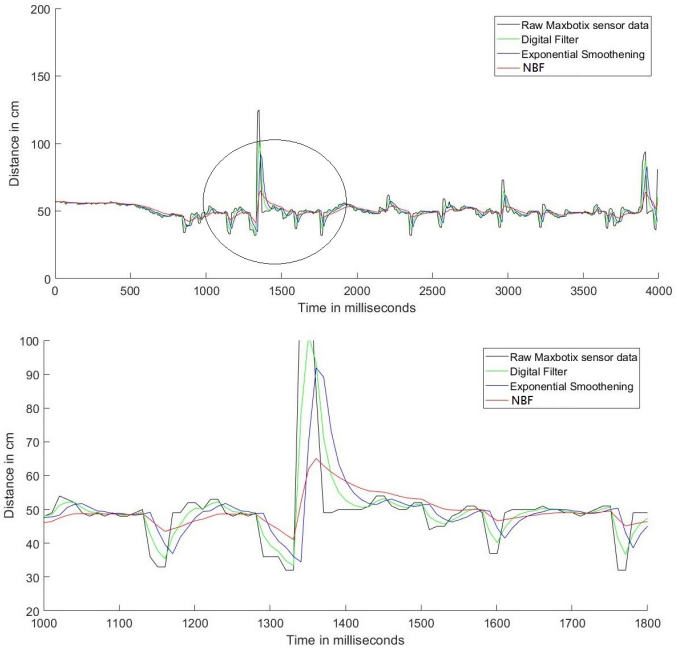

Fig. 4. Comparison between raw data and various filtered data for the Maxbotix MB1220 with blown-up view on the bottom. The bottom figure clearly shows capability of the NBF to substantially reduce deviation from the 'true' value of proximity.

Since the emphasis was on safe smooth movement of the quadrotor,the NBF was chosen as the ideal filter for the task.

4.3 Sensor Fusion

As was observed from the comparison of the sensors in Table 2:

(1) The Maxbotix sensor has a good measuring angle for small moving objects and hence is ideal for detecting such objects at moderate to large distances (greater than 40 cm). However, it has a minimum reading value of 20 cm and is very noisy and unreliable when the actual distance is close to 20 cm.
(2) The HC-SR04 sensor is poor in sensing moving objects at large distances. However, it performs well for distances smaller than 30 cm for almost any kind of object. It also has a minimum reading value of 3 cm.

Hence, it was decided to employ sensor fusion using sigmoid functions. This function assigns a value between 0 and 1 to the variable α.

$$\alpha = \frac{1}{1 + e^{-k(x - x_{ref})}} \tag{11}$$

Here, the parameters k and x_{ref} were chosen as 0.6 and 30 respectively for smooth transition starting at x=20 and ending at x=40. These parameters were chosen keeping in mind sensor characteristics and comparisons as highlighted in Table 4.1. This fusion technique's efficacy was confirmed in a hardware implementation test wherein the two sensors mounted together, operating in fused mode, were moved closer and away from a rough cardboard panel while

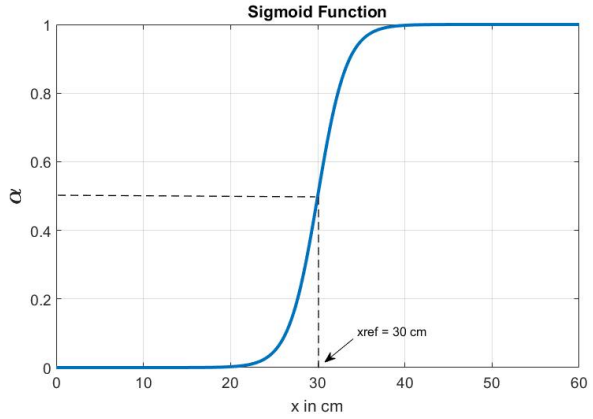

Fig. 5. Sigmoid function with k=0.6 and $x_{ref} = 30\,cm$

simultaneously shaking and disturbing their orientation to simulate attitude disturbances on a quadrotor.

$$d_F = \alpha d_M + (1 - \alpha) d_H \tag{12}$$

where d_F is the fused value and d_M and d_H are the individually filtered values of the Maxbotix and the HC-SR04 sensors respectively.

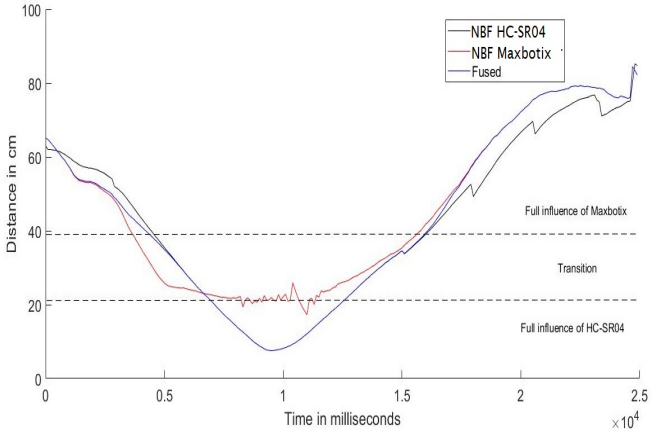

Fig. 6. Test results of sensor fusion output in comparison with individual filtered sensor data

As is evident from the plot, this method proves to be effective for the reasons given below:

(1) Since the output completely follows the output of the HC-SR04 below 20 cm, noise and out-of-range behaviour exhibited by the Maxbotix sensor is ignored for this range.
(2) At larger distances (greater than 30 cm), the increasing share of the Maxbotix sensor reading in the output ensures that the noisy HC-SR04 data is neglected.

5. CONTROLLER DESIGN AND SIMULATION

The equation of motion of the object with respect to ground is given by $y(t)$. One of the main goals of this project is enabling the quadrotor to follow a human; hence, to simulate a person walking at normal speed of about $140\,\mathrm{cm\,s^{-1}}$ (Levine and Norenzayan (1999)), the equation used is

$$y(t) = 140t \tag{13}$$

For a person moving back and forth, the equation of motion may not be as straightforward. Hence, to simulate such a situation, a sinusoidal input for amplitude 50 cm and time period of 3 seconds has been used.

$$y(t) = 50\sin(\frac{2\pi t}{3}) \tag{14}$$

5.1 PID Controller

The controller used a PID controller. A reference value of 75 cm between the object and the quadrotor is selected to be regulated by the controller. The model is given by

$$U_p(t) = k_p e(t) + k_d \frac{de(t)}{dt} + k_i \int_0^t e(t) \tag{15}$$

where $e(t)$ is the difference between the observed distance and the reference distance. The reference in the implementation is selected as 75 cm. k_p, k_i and k_d are the P,I and D gains respectively. $U_p(t)$ is the controller output in terms of PWM signal value limited between +65 and -65 to limit pitch angle to a maximum of 12°.

The algorithm runs the control loop only when commanded to do so (from the transmitter) and when the distance measured is within 35 cm of the reference distance. At all other times, it is under manual control.

5.2 SIMULINK® Model

The control system along with the two aforementioned probable motions of the object (person) was modelled on SIMULINK®. The reference input is set at 35 cm as detection begins at this distance the the reference in the SIMULINK model is left at 0 cm for simplicity. The response will remain the same as when the reference is kept at 75 cm with a detection at either $75 + 35 = 110$ cm or $75 - 35 = 40$ cm.

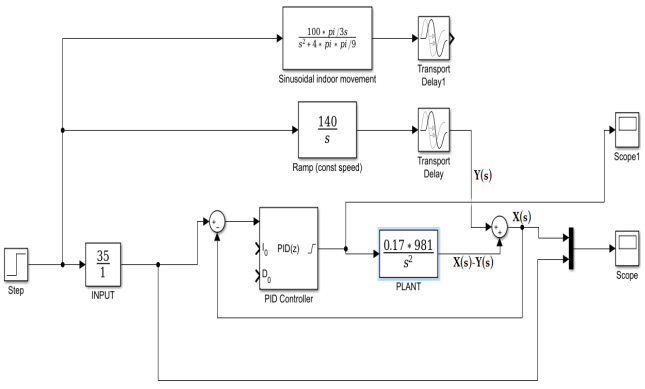

Fig. 7. SIMULINK® Model on MATLAB® with the linear walking feedforward component connected and sinusoidal back and forth available for connection on top

The system was tuned with the assistance of the *PID Tuner Toolbox* in MATLAB and simulation results (8 seconds) for $k_p = 3.5, k_i = 0$ and $k_d = 0.6$ are plotted below.

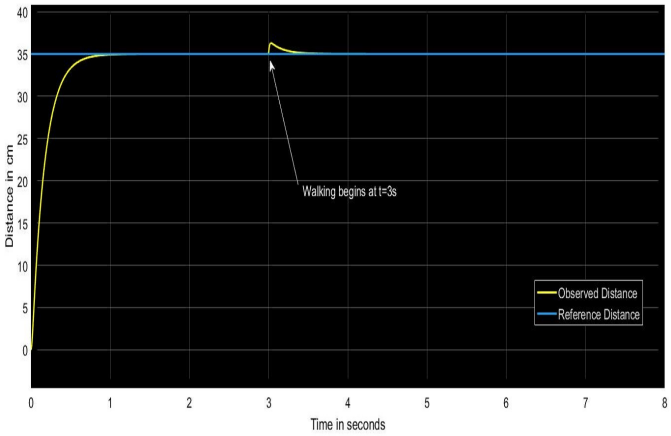

Fig. 8. Response for Stationary and then Linear Walking

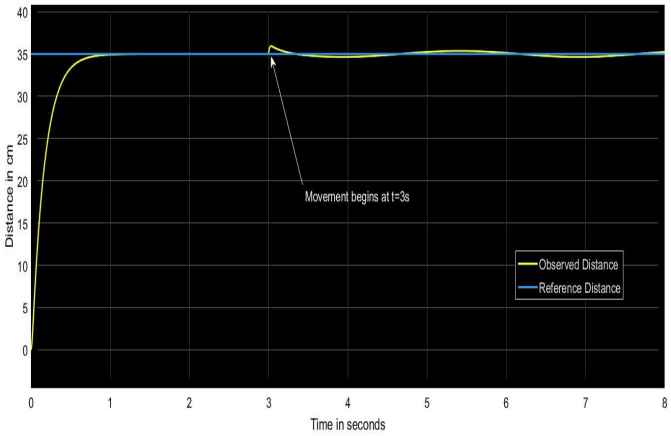

Fig. 9. Response for Stationary and then Back and Forth Movement of Person

The results from Fig. 8 and Fig. 9 show that the quadrotor settles at the desired relative distance from the object within 1 second and does not deviate by more than 2 cm from its desired position once motion begins. In the case of sinusoidal motion, there exists a persistent sinusoidally varying error due to the feedforward component in the system model. However, its amplitude is less than 1 cm. Therefore, the controller performance can be said to be satisfactory.

6. IMPLEMENTATION AND RESULTS

The controller was implemented on Arduino UNO (16 MHz processing speed) with three modes:

(1) *Stabilise Mode*: In this mode, APM performs attitude stabilization but apart from that, the yaw, pitch, roll and thrust inputs lie in the control of the transmitter held by the user.
(2) *Altitude Hold Mode*: In this mode, the thrust control is almost completely taken away from the user and is controlled by APM to maintain constant altitude using data from the on-board barometer; rest of the

control is retained by user as in Stabilise Mode. Channel 5 of the transmitter is used to activate this mode.

(3) *'Follow Me' Mode*: This is the mode for which the entire control system was developed. In this mode, the quadrotor scans for an object in front of it and follows it when the specified distance criterion is satisfied. It is activated within the Altitude Hold mode. The object detection begins when Channel 6 of the transmitter is set to a value above a give threshold.

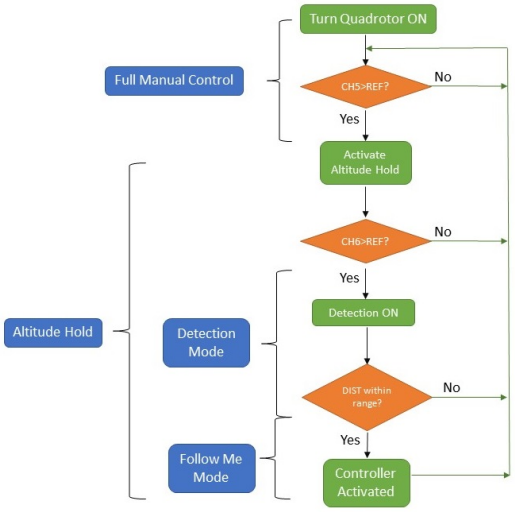

Fig. 10. Loop Flow Chart of Implemented Algorithm on the Microcontroller

The 'object' used was a panel of rough cardboard mounted on top of a ground robot capable of translating linearly and sinusoidally in one dimension. The tests were carried out by activating the quadrotor's 'follow me' control loop at the desired distance from the object (ground robot) while the object tracked a linear and sinusoidal trajectory in between sufficient dwell time. Position data of the ground robot was gathered from its motor encoders and the same from the quadrotor was gathered by logging and integrating its accelerometer data. The maximum and mean absolute tracking error during the linear walking experiment was about 55 cm and 13 cm respectively. The error was not more than 20 cm during the sinusoidal back and forth movement testing. In spite of the drift from the accelerometer data integration causing a noticeable error in the experimental data, the performance of the controller as seen from Fig. 11 and Fig. 12 is satisfactory for the application. Hence, the entire system was validated.

7. CONCLUSION

In this paper, a cheap (both computationally and financially), novel and effective way of object detection and tracking in one dimension was demonstrated.The entire system including the quadrotor kit was assembled in under $150 requiring only a 16 MHz processor. It is easy to see how this system can be extended to account for multiple dimensions. Such systems can find applications in both indoor and outdoor localization with respect to fixed and

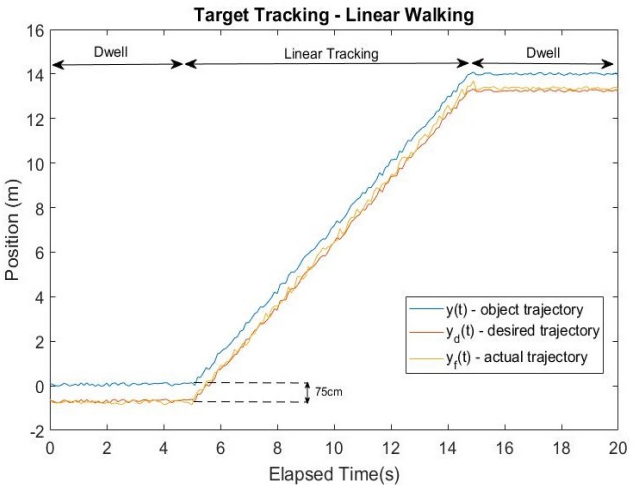

Fig. 11. Experimental Results: Target Tracking of a Linearly Moving Object

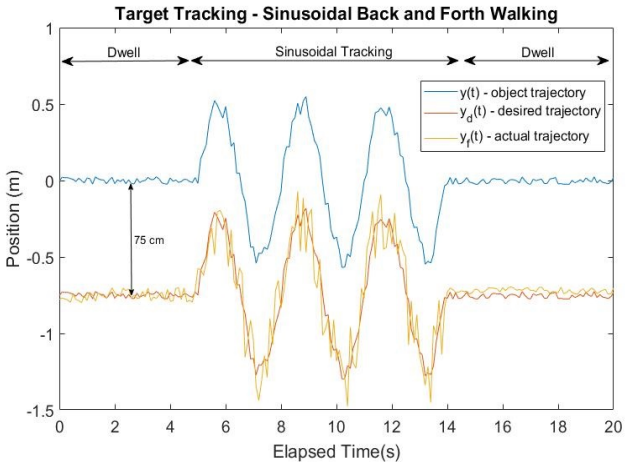

Fig. 12. Experimental Results: Target Tracking of a Sinusoidally Moving Object

moving references. Swarm aerial robots which require coordinated movement of drones may utilize this system in the absence of GPS, sufficient illumination or other means of localization.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the *Department of Aerospace Engineering, Indian Institute of Technology Madras* for sponsoring this work during my time there as a summer fellow in 2017. I would also like to thank my guide, *Professor Joel George*, for his support and valuable inputs.

REFERENCES

Argentim, L.M., Rezende, W.C., Santos, P.E., and Aguiar, R.A. (2013). Pid, lqr and lqr-pid on a quadcopter platform. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 1–6. doi: 10.1109/ICIEV.2013.6572698.

Becker, M., Sampaio, R.C.B., Bouabdallah, S., Perrot, V.d., and Siegwart, R. (2012). In-flight collision avoidance controller based only on os4 embedded sensors.

Journal of the Brazilian Society of Mechanical Sciences and Engineering, 34(3), 294–307.

Blösch, M., Weiss, S., Scaramuzza, D., and Siegwart, R. (2010). Vision based mav navigation in unknown and unstructured environments. In *Robotics and automation (ICRA), 2010 IEEE international conference on*, 21–28. IEEE.

Bouabdallah, S. and Siegwart, R. (2007). Full control of a quadrotor. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 153–158. doi:10.1109/IROS.2007.4399042.

Boudjit, K. and Larbes, C. (2015). Detection and implementation autonomous target tracking with a quadrotor ar.drone. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 02, 223–230.

Celik, K., Chung, S.J., Clausman, M., and Somani, A.K. (2009). Monocular vision slam for indoor aerial vehicles. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1566–1573. doi:10.1109/IROS.2009.5354050.

Dang, C.T., Pham, H.T., Pham, T.B., and Truong, N.V. (2013). Vision based ground object tracking using ar.drone quadrotor. In *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 146–151. doi:10.1109/ICCAIS.2013.6720545.

Engel, J., Sturm, J., and Cremers, D. (2012a). Accurate figure flying with a quadrocopter using onboard visual and inertial sensing. *Imu*, 320, 240.

Engel, J., Sturm, J., and Cremers, D. (2012b). Camera-based navigation of a low-cost quadrocopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2815–2821. IEEE.

Fang, Z. and Gao, W. (2011). Adaptive integral backstepping control of a micro-quadrotor. In *2011 2nd International Conference on Intelligent Control and Information Processing*, volume 2, 910–915. doi:10.1109/ICICIP.2011.6008382.

Gageik, N., Benz, P., and Montenegro, S. (2015). Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3, 599–609. doi:10.1109/ACCESS.2015.2432455.

Jurado, F., Palacios, G., and Flores, F. (2012). Vision-based trajectory tracking on the 3d virtual space for a quadrotor. In *2012 IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference*, 31–36. doi:10.1109/CERMA.2012.13.

Kendall, A.G., Salvapantula, N.N., and Stol, K.A. (2014). On-board object tracking control of a quadcopter with monocular vision. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 404–411. doi:10.1109/ICUAS.2014.6842280.

Levine, R.V. and Norenzayan, A. (1999). The pace of life in 31 countries. *Journal of Cross-Cultural Psychology*, 30(2), 178–205. doi:10.1177/0022022199030002003. URL `https://doi.org/10.1177/0022022199030002003`.

Li, J. and Li, Y. (2011). Dynamic analysis and pid control for a quadrotor. In *2011 IEEE International Conference on Mechatronics and Automation*, 573–578. doi:10.1109/ICMA.2011.5985724.

Mth, K., Buoniu, L., Barabs, L., Iuga, C.I., Miclea, L., and Braband, J. (2016). Vision-based control of a quadrotor for an object inspection scenario. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 849–857. doi:10.1109/ICUAS.2016.7502522.

Salih, A.L., Moghavvemi, M., Mohamed, H.A.F., and Gaeid, K.S. (2010). Modelling and pid controller design for a quadrotor unmanned air vehicle. In *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, volume 1, 1–5. doi:10.1109/AQTR.2010.5520914.

Scherer, S., Singh, S., Chamberlain, L., and Elgersma, M. (2008). Flying fast and low among obstacles: Methodology and experiments. *Int. J. Rob. Res.*, 27(5), 549–574. doi:10.1177/0278364908090949. URL `http://dx.doi.org/10.1177/0278364908090949`.

Shen, S., Michael, N., and Kumar, V. (2011). Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle. In *2011 IEEE International Conference on Robotics and Automation*, 2968–2969. doi:10.1109/ICRA.2011.5980364.

Teulire, C., Eck, L., and Marchand, E. (2011). Chasing a moving target from a flying uav. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4929–4934. doi:10.1109/IROS.2011.6094404.

Tian, C., Wang, J., Yin, Z., and Yu, G. (2016). Integral backstepping based nonlinear control for quadrotor. In *2016 35th Chinese Control Conference (CCC)*, 10581–10585. doi:10.1109/ChiCC.2016.7555034.

.

Preprint submitted to Third IFAC International Conference on Advances in Control and Optimization of Dynamical Systems. Received September 30, 2017.