Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-1 (2018) 154–159

Target Tracking by a Quadrotor Using Proximity Sensor Fusion Based on a Sigmoid Function [⋆]

Varun Nayak [*] **Raksha Rao Karaya** [**]

[*] *Department of Mechanical Engineering, Birla Institute of Technology and Science, Pilani - K K Birla Goa Campus (e-mail: f20140086@goa.bits-pilani.ac.in).*
[**] *Department of Aerospace Engineering, 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 experimental verification of a one-dimensional object tracking quadrotor that can detect and follow a solid object in front of it by regulating its distance from the object. A 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. This system uses a Proportional-Integral-Derivative (PID) controller to generate a single high-level pitch reference based on the sensor fusion output, in order to track a target. Low-level attitude control and altitude maintenance is simultaneously performed by a commercially available autopilot system.

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. Nowadays, there is also an increasing trend in the use of micro-class UAVs for consumer applications. 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 their surrounding environment.

The primary problem of controlling a quadrotor involves state estimation and motor control for maintaining desired attitude and thrust. Many researchers have solved this problem by employing different control techniques – from classical PID or LQR control to more complex nonlinear integral backstepping control. More complicated approaches are taken to improve robustness by including gyroscopic and aerodynamic effects. 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.

Problems involving detection, avoidance and tracking are fundamental to autonomy. Boudjit and Larbes (2015) and Dang et al. (2013) have developed an 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.

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.

In this paper, a 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

[⋆] 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.

10.1016/j.ifacol.2018.05.026

sensing is performed using two ultrasonic sensors with distinct characteristics. The reliability of the proximity data is vastly improved by a combination of a noise-based filter (NBF) and an experimentally derived sensor fusion technique based on a sigmoid function. This computationally inexpensive method was developed and tested using a low-cost microcontroller, entirely on board the quadrotor. Since this scheme depends only on ultrasound signals, it remains effective even in poorly lit environments.

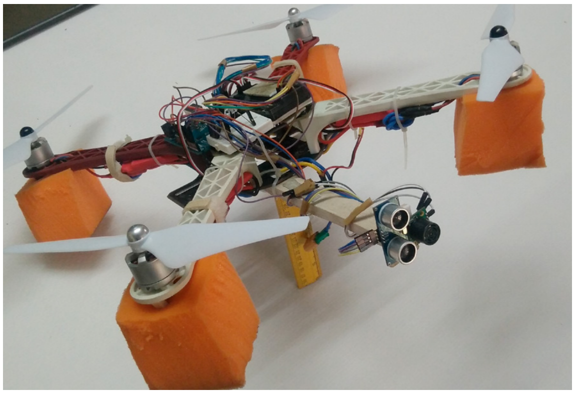

Fig. 1. The fully assembled quadrotor with two front-mounted proximity sensors.

2. QUADROTOR PLATFORM AND SENSOR SUITE

The quadrotor platform used in the development of this system is the DJI F-450. 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 DJI F-450 kit.

In addition, the quadrotor is equipped with an APM 2.5 [1] for low-level control as well as an Arduino® UNO [2] 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 can maintain altitude using an on board barometer. Hence, the quadrotor as shown in Fig. 2 is assumed to be static in the vertical direction. The analysis that follows is a simplified model with the objective of aiding controller design.

As shown in Fig. 2, the coordinate frame attached to the ground measures the distance of the object y from its origin. Another frame attached to the object measures the distance of the quadrotor x from the object. Applying Newton's 2^{nd} law on the system in the vertical and horizontal directions respectively, we obtain the following equations:

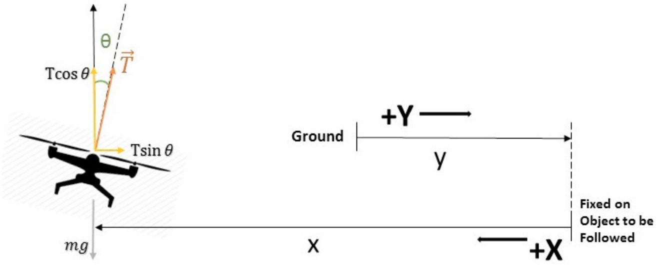

Fig. 2. System dynamics of a quadrotor tracking a moving object in one dimension.

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

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

Here, $\theta(t)$ 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 (as the pitch angle shall be limited to 15°), we can develop the above dynamic equations into an 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 obtain the following:

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

Here, $P(s)$ 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

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 additional tests.

[1] http://ardupilot.org/copter/

[2] https://www.arduino.cc/

[3] This is determined using the Mission Planner client software for APM (http://ardupilot.org/dev/docs/apmcopter-programming-attitude-control-2.html).

The two ultrasonic proximity sensors used are the HC-SR04 and the Maxbotix® MB1220. Both are cheap and easy to integrate. From the comparison given in Table 2, it is evident that the two sensors have many distinct characteristics. This led to the decision of employing sensor fusion based on a sigmoid function.

4.2 Data Filtering

Since a stable, reliable and accurate stream of distance data is required as an input to the controller, the sensor data was processed through an effective filter. A comparative analysis of three filters – exponential smoothing, digital filtering and noise-based filtering was done through implementation on hardware in order to choose the best 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.

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

Here, 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.

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

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 reading the sensor at specific time intervals, depending on its update frequency.

Noise-Based Filter (NBF): The NBF designed specifically for this system recursively calculates the estimate of distance combining prior estimates, the sensor model and noisy measurements. The scalar variable distance is measured by the sensor, whose best estimate is stored in a variable labeled as the output of the filter. This is essentially a low-pass filter with a varying time constant based on a Gaussian model. A gain $G[n]$ distributes weight between the previous estimate and the current sensor reading. At the nth time step, the gain is updated using the sensor variance σ_s^2 (known from intrinsic sensor characteristics) and the estimate variance $\sigma_x^2[n]$.

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

Next, the new estimate $x[n]$ is calculated using the previous estimate $x[n-1]$ and the current raw sensor reading.

$$x[n] = (1 - G[n])x[n-1] + G[n]S[n] \quad (8)$$

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

$$\sigma_x^2[n] = (1 - G[n])\sigma_x^2[n-1] \quad (9)$$

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

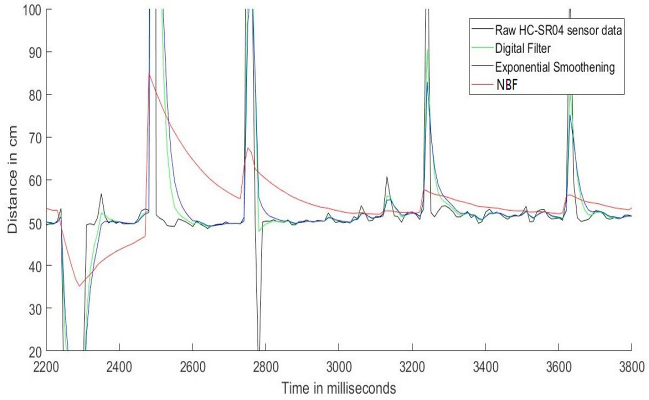

Fig. 3. Comparison between raw data and various filtered data for the HC-SR04 sensor: The figure clearly shows the capability of the NBF to substantially reduce deviation from the 'true' value of proximity.

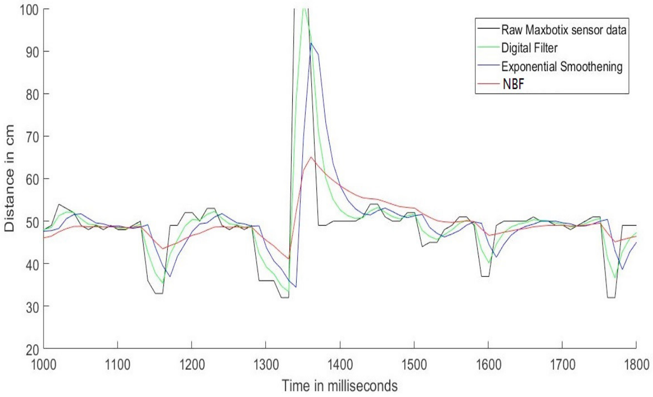

Fig. 4. Comparison between raw data and various filtered data for the Maxbotix MB1220: The figure clearly shows the capability of the NBF to substantially reduce deviation from the 'true' value of proximity.

From Fig. 3 and Fig. 4, it can be seen that the outputs 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.

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

Since the emphasis was on safe and 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 has a minimum reading value of 3 cm.

These differences motivated the decision to employ sensor fusion using a sigmoid function. This function assigns a value between 0 and 1 to the variable α.

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

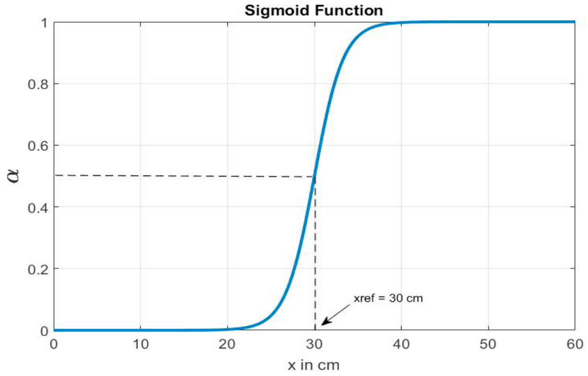

Fig. 5. Sigmoid function with k=0.6 and $x_{ref} = 30\,cm$.

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 simultaneously shaking and disturbing their orientation to simulate attitude disturbances on a quadrotor. The fusion step obeys the following equation:

$$d_F = \alpha\, d_M + (1 - \alpha)\, d_H \tag{11}$$

Here, d_F is the fused value (output) and d_M and d_H are the individually filtered values of the Maxbotix and the HC-SR04 sensors respectively.

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 work is to enable the quadrotor to follow a human. Hence, to simulate a person walking at normal speed of about 140 cm s^{-1} (Levine and Norenzayan (1999)), the equation used is as follows:

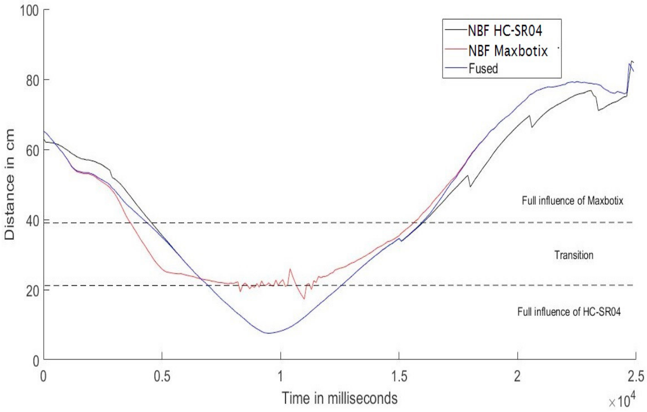

Fig. 6. Test results of sensor fusion output in comparison with individual sensor data.

$$y(t) = 140\, t \tag{12}$$

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.

$$y(t) = 50 \sin(\frac{2\pi t}{3}) \tag{13}$$

5.1 PID Controller

The controller uses a PID control law for the regulation of distance between the object and the quadrotor. A reference value of 75 cm between the object and the quadrotor was selected.

$$U_p(t) = k_p\, e(t) + k_d \frac{de(t)}{dt} + k_i \int_0^t e(t) \tag{14}$$

Here, $e(t)$ 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 the PWM signal value constrained 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.

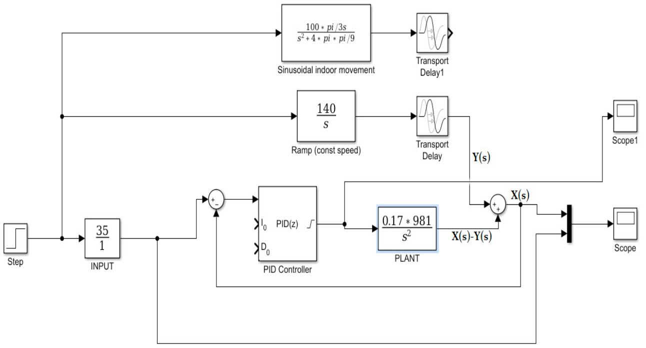

Fig. 7. SIMULINK® model of the closed-loop system. It shows two possible types of motions - linear and sinusoidal - as feedforward inputs.

The simulated system response (8 seconds) for $k_p = 3.5, k_i = 0$ and $k_d = 0.6$ were plotted as shown in Fig.8 and Fig.9. The integral term remains at zero as this prevents the feedforward component in the system from generating an unstable closed-loop response.

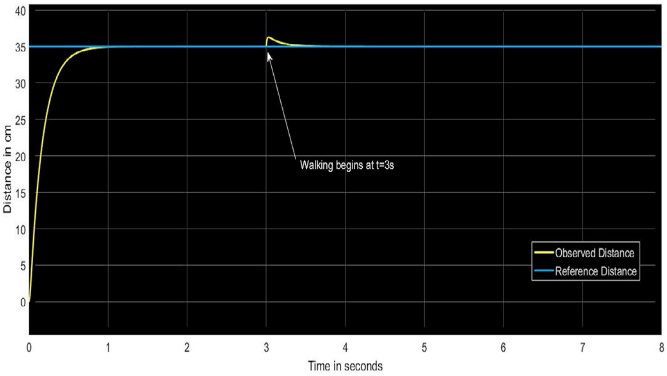

Fig. 8. Response for linear movement of the object.

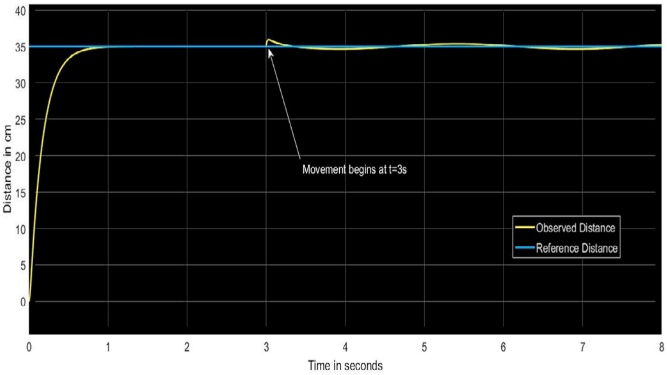

Fig. 9. Response for sinusoidal movement of the object.

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 an Arduino UNO (16 MHz processing speed) with three modes:

(1) *Stabilise Mode*: In this mode, APM performs attitude stabilization. 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. This mode is nested in altitude hold. The object detection begins when Channel 6 of the transmitter is set to a value above a given threshold.

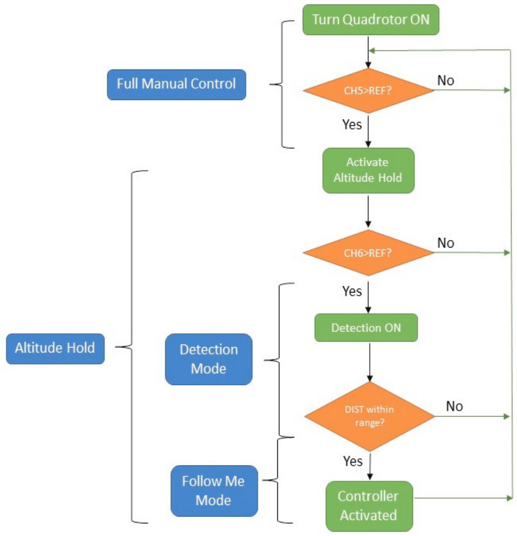

Fig. 10. Loop flow chart of the algorithm.

The 'object' used was a cardboard panel mounted on a ground robot. The tests were carried out by activating the quadrotor's 'follow me' control loop at a preset distance from the object while it traversed linear and sinusoidal trajectories in between sufficient dwell times. The trajectory of the ground robot was gathered from its motor encoders and the same from the quadrotor was gathered from its IMU. 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 tracking. In spite of the drift from the accelerometer data integration causing a noticeable error in the experimental results, the performance of the controller as seen from Fig. 11 and Fig. 12 is satisfactory for this application.

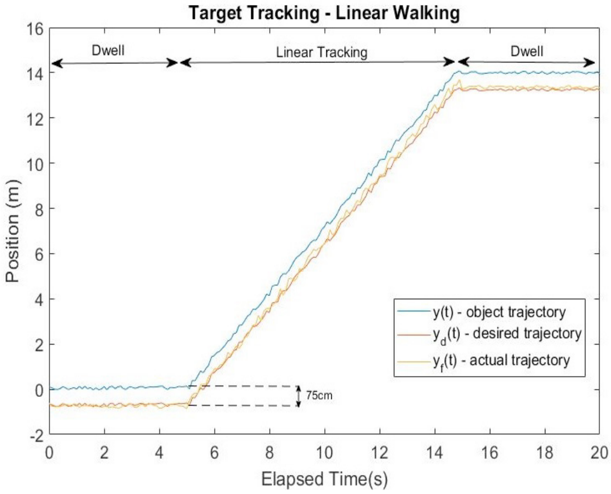

Fig. 11. Experimental Results: Target tracking of a linearly moving object.

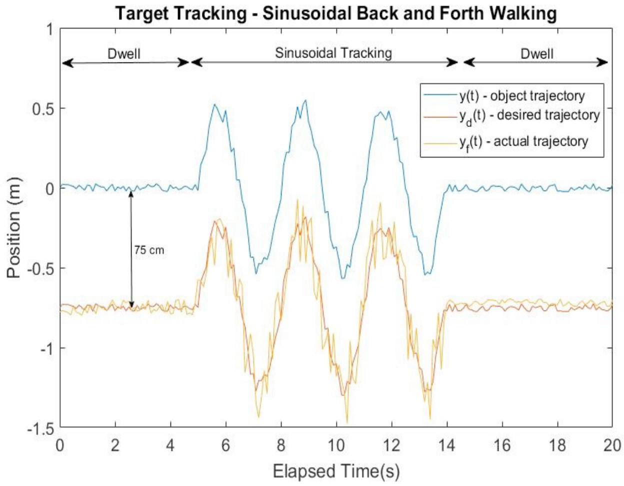

Fig. 12. Experimental Results: Target tracking of a sinusoidally moving object.

7. CONCLUSION

In this paper, a computationally cheap 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 on board 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 moving references. Robot swarms requiring coordinated movement may utilize this system in GPS-denied or visually degraded environments.

REFERENCES

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.

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.

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`.

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.

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.