Indoor Navigation of Quadrotors via Ultra-Wideband Wireless Technology

Conference Paper · November 2018

DOI: 10.1109/RTUWO.2018.8587889

3 authors:

Ilias Papastratis
The Centre for Research and Technology, Hellas
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Themistoklis Charalambous
Aalto University
130 PUBLICATIONS **1,382** CITATIONS
SEE PROFILE

Nikolaos Pappas
Linköping University
117 PUBLICATIONS **812** CITATIONS
SEE PROFILE

Indoor Navigation of Quadrotors via Ultra-Wideband Wireless Technology

Ilias Papastratis, Themistoklis Charalambous, and Nikolaos Pappas

***Abstract*—This work focuses on controlling the position and trajectory of a quadrotor indoors with high positioning accuracy. The movement of the quadrotor is tracked by an Ultra-WideBand (UWB) system which provides wireless message communication that enables communication with the quadrotor. In this feedback control system, the main challenge is that the system has limited communication bandwidth, which may cause packet losses and time delays, and the quadrotor is extremely sensitive to such effects. Our main objective is to develop a controller and jointly tune a positioning system to stabilize the system by trading between latency and reliability. The performance of our developed schemes are verified using a testbed platform.**

I. Introduction

In the last recent years, unmanned aerial vehicles (UAVs), such as quadrotors, have become very popular due to the wide variety of applications, such as, surveillance, inspection, industrial tasks, delivery, agriculture, and data acquisition. Most UAVs use GPS-based positioning system which is suitable only for outdoor environment localization. Moreover, the precision of satellite navigation is very limited (position accuracy is approximately 1 meter), rendering it not suitable for high-precision tasks, and it does not work indoors. Quadrotors, however, are used nowadays for several indoor applications, such as, object transportation, inspection and delivering parts. Because of these and other foreseeable applications, research in indoor localization systems for UAVs, such as quadrotors is becoming a necessity.

In order to improve the accuracy, different methods have been developed. Some of these improve the satellite accuracy by means of data fusion with inertial navigation systems (INS) [1]. In many cases positioning methods are based on computer vision [2], [3] or a combination of INS and Vision [4]. Motion capture systems are also used for autonomous flights [5]. However, vision-based approaches are very expensive and require a wired infrastructure. Cameras must be calibrated and more processing power is required for computer vision. They provide high precision and update rate, but they are expensive, difficult to set up, and add more payload to the quadrotor. To overcome the shortcomings of the aforementioned technologies, in this work we focus on the deployment of an UWB localization system for controlling the navigation of a quadrotor. Note that different position estimation methods have been implemented using UWB measurements [6]. Other works in the literature have also considered similar technologies; for example, a similar setup implemented the extended Kalman filter (EKF) [7], UWB measurements with accelerometer and gyroscope measurements were fused in [8], a mobile UWB anchor was utilized to improve the positioning accuracy [9], [10]. GPS emulation has also been implemented using UWB system [11]. Vision-based odometry has also been fused with UWB measurements to improve accuracy [12].

The difference between our work to that of aforementioned methods is that we focus on the positioning accuracy and the interplay between communication and control. More specifically, we control the position of the quadrotor using the UWB measurements for different communication settings. To evaluate the performance of our setup and controller we used two tasks: the first is to make the quadrotor hover at a desired position, and the second is to make the quadrotor follow certain trajectories.

II. Preliminaries

A. UWB technology for localization

UWB localization is a radio technology characterized by its very large bandwidth compared to conventional narrow-band systems, and in particular features high positioning accuracy (due to a time resolution in the order of nanoseconds), and high material penetrability. A UWB tag transmits a periodic data request to all anchors within its range. The tag measures response time by utilizing the measure the time of flight of the RF signals and is able to determine distance to each anchor. This information is used to determine the tag's location with respect to the anchors' positions via *multilateration*; see, e.g., [13]. Note that autocalibration is possible and the position of the anchors is not necessary to be known *a priori*. The difficulty of this approach in our setup lies in the fact that, despite the measurements being noisy, the distances between an anchor and a tag are retrieved in a cyclic fashion, thus introducing localization error, especially when the quadrotor is moving at a high speed.

B. Hardware setup

The hardware platform is composed of a quadcopter, an onboard computer that communicates with the onboard controller of the quadcopter, the UWB communication system and a router to establish the communication between the personal computer and the onboard computer; see Fig. 1.

I. Papastratis and T. Charalambous are with the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland (E-mails: name.surname@aalto.fi).

N. Pappas is with the Department of Science and Technology, Linköping University, Campus Norrköping, Sweden (E-mail: nikolaos.pappas@liu.se).

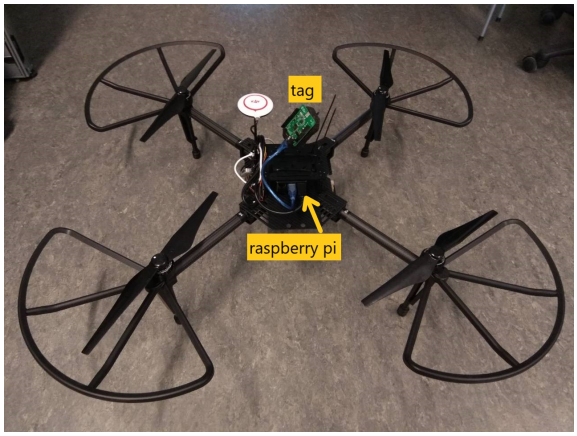

Fig. 1. Quadcopter Build. It consists of a DJI Matrice 100 quadcopter, a Raspberry Pi 3 onboard computer, and a Pozyx tag..

The *DJI Matrice 100 quadcopter* is a fully programmable platform that can be programmed using the DJI Software Development Kit (DJI SDK). The platform is able to communicate with a variety of components such as camera, ultrasonic or UWB sensors, GPS, and an onboard computer.

The *Raspberry Pi 3 model B* is used as the onboard computer that communicates with the flight controller of the quadrotor through a serial USB cable. It has a quad core 1.2 GHz processor which is capable of doing this task efficiently. There is also a WiFi module for wireless communication and USB ports which are used to connect to the UWB sensor and the flight controller. Raspberry receives the data from the UWB sensor, calculates the desired commands and sends them to the flight controller. The raspberry is equipped with Ubuntu 16.04 operating system.

The *positioning system* consists of seven modules, of which six anchors (see Fig. 2, right) and one tag (see Fig. 2, left). Anchors are placed in steady position in the indoor space. The tag is connected through USB to the onboard computer.

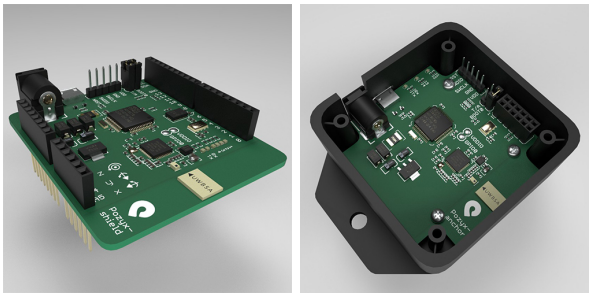

Fig. 2. *Left:* The Pozyx tag communicates and finds its distance from each anchor and then calculates the position through multilateration. *Right:* Pozyx anchor.

Positioning system configuration. We use the developer's kit from Pozyx. The modules are equipped with the decawave DW1000 UWB transceiver. The range of DW1000 is limited to $100m$ or they can cover a maximum area of $35m \times 35m$. The range is smaller than other UWB systems but Pozyx devices can achieve a higher refresh rate which is important for the purpose of this project. Pozyx modules with DW1000 are available in a form of standalone devices and Arduino add-on modules. Pozyx devices can also be configured with different settings to provide better accuracy, different refresh rates or area coverage according to the project needs. The settings that can be set are:

1) *Channel:* We can select from six independent channels for communication. All devices must have the same UWB channel in order to communicate. Usually lower frequencies result in an increased communication range.
2) *Bitrate:* We can choose three different values: 110 Kbits/sec, 850 Kbits/sec, and 6.81 Mbits/sec. A higher bitrate will result in short messages and faster communication but it will reduce the operating range of the sensors.
3) *Pulse Repetition frequency:* Two settings are possible for the pulse repetition frequency, 16 MHz, and 64 MHz.
4) *Preamble length:* This setting has eight different options 64, 128, 256, 512, 1024, 1536, 2048, and 4096 symbols. A shorter length can be set for faster communication but the operating range will be shorter. However, if a longer operating range is required then a length with more symbols should be used.

The anchors should be placed high and in line of sight of the quadrotor. It is preferred to spread the anchors in a rectangular shape and the distances between them depend on the area we want to cover and the UWB settings.

III. Quadcopter Model

The model used in this paper follows that of [14] and [15]. The quadcopter is actuated by four rotors (see Fig. 3), each consisting of a propeller fitted to a separately powered DC motor. Two of the rotors rotate clockwise, while the other two rotate counter-clockwise to cancel the yawing moment; see, e.g., [16]. All the movements are generated by varying the angular speed of the rotors, denoted by $(\Omega_1, \Omega_2, \Omega_3, \Omega_4)$.

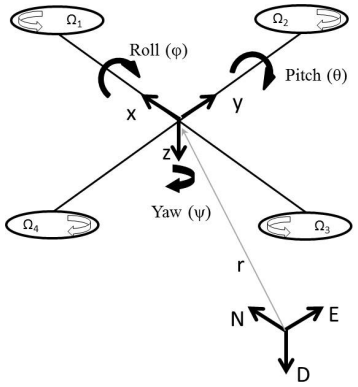

Fig. 3. Quadrotor reference frames [15].

The quadrotor is a 6 degrees-of-freedom object, thus 6 variables are used to express its position in space (x, y, z, ϕ, θ and ψ). x, y and z represent the coordinates of the quadrotor's center of mass from an Earth inertial frame. ϕ, θ and ψ are the three Euler angles representing the orientation of the quadrotor: ϕ is called the roll angle which is the angle about the x-axis, θ is the pitch angle about the y-axis, and ψ is the yaw angle about the z-axis (see Fig. 3).

$$R(\phi,\theta,\psi)=\begin{bmatrix}\cos\psi\cos\theta & \cos\psi\sin\phi\sin\theta-\cos\phi\sin\psi & \sin\phi\sin\psi+\cos\phi\sin\theta\\ \cos\theta\sin\psi & \cos\phi\cos\phi+\sin\phi\sin\psi+\sin\theta & \cos\phi\sin\psi\sin\theta-\cos\psi\sin\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta\end{bmatrix} \tag{1}$$

A. Kinematics Model

The Earth inertial frame uses the N-E-D notation where the axes point to the North, East and Downwards, respectively (see Fig. 3). The orientation of the quadrotor is described using roll (ϕ), pitch (θ) and yaw (ψ) angles representing rotations about the x, y and z-axes respectively. The rotation $R(\phi,\theta,\psi)$ from the body frame to the inertial frame describes the orientation of the quadrotor and it is given by equation (1); this is the result of three independent rotations cascaded through matrix multiplication, whose order (order matters) is yaw (ψ) followed by pitch (θ) followed by roll (ϕ) around the z, y and x axes, respectively. The rotation matrix $R(\phi,\theta,\psi)$ is needed when formulating the dynamics model (see § III-B) because some states are measured in the body frame (e.g., the forces produced by the rotors) while some others are measured in the inertial frame (e.g., the gravitational forces and the quadrotor's absolute position), and therefore a transformation from one frame to the other is necessitated.

B. Dynamics Model

1) Rotational Equation of Motion: The dynamics model [15] consists of the rotational and translational motions. The rotational equations of motion are derived in the body frame using the Newton-Euler method as follows

$$J\dot{\omega}+\omega\times J\omega+M_G=M_B, \tag{2}$$

where J is the quadrotor's diagonal inertia matrix, ω is the angular body rate, $J\dot{\omega}$ and $\omega\times J\omega$ represent the rate of change of angular momentum in the body frame. M_G represents the gyroscopic moments due to the rotor's inertia and is given by

$$M_G:=\omega\times\begin{bmatrix}0 & 0 & J_r\Omega_r\end{bmatrix}^T, \tag{3}$$

where J_r represents rotors inertia and $\Omega_r:=-\Omega_1+\Omega_2-\Omega_3+\Omega_4$ is rotor's relative speed. As an effect of rotation, there is a generated force $F_i=k_F\Omega_i^2$, called the aerodynamic force or the lift force, and there is a generated moment $M_i=k_M\Omega_i^2$, called the aerodynamic moment [15]; k_F and k_M are the aerodynamic force and moment constants, respectively. The moments acting on the quadrotor in the body frame are given by

$$M_B=\begin{bmatrix}lk_F(-\Omega_2^2+\Omega_4^2),\\ lk_F(\Omega_1^2-\Omega_3^2),\\ k_M(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2)\end{bmatrix}, \tag{4}$$

where l is the moment arm, the distance from the axis of rotation of the rotors to the center of the quadrotor.

2) Translational Equation of Motion: The translational equations of motion are derived in a North-East-Down navigation frame using Newton's second law [14]

$$m\ddot{r}=\begin{bmatrix}0 & 0 & mg\end{bmatrix}^T+R(\phi,\theta,\psi)F_B, \tag{5}$$

where $r=\begin{bmatrix}x & y & z\end{bmatrix}^T$ is the quadrotor's position in the navigation frame, m is the quadrotor's mass and g is the acceleration due to gravity.

The nongravitational forces, F_B, acting on the quadrotor in the body frame are given by

$$F_B=\begin{bmatrix}0\\0\\k_F(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2)\end{bmatrix}=:\begin{bmatrix}0\\0\\-U_1\end{bmatrix}. \tag{6}$$

IV. Position Control Design

Unlike the altitude and orientation of the quadrotor, its 2D position, given by coordinates x and y, is not decoupled. Hence, the x and y coordinates can be controlled through the roll and pitch angles. The desired roll and pitch angles ϕ_d and θ_d can be calculated from the translational equations of motion, if we rewrite it in terms of accelerations [14], i.e.,

$$\ddot{x}=\frac{-U_1}{m}(\sin\phi_d\sin\psi+\cos\phi_d\sin\theta_d\cos\psi), \tag{7a}$$

$$\ddot{y}=\frac{-U_1}{m}(\cos\phi_d\sin\theta_d\sin\psi-\sin\phi_d\cos\psi). \tag{7b}$$

Since the quadrotor is operating around hover position and the angles are small we can simplify the equations using small angle assumption, i.e., $\sin\phi_d\approx\phi_d$, $\sin\theta_d\approx\theta_d$, $\cos\phi_d\approx 1$, and $\cos\theta_d\approx 1$. Hence, the new equations are

$$\ddot{x}=\frac{-U_1}{m}(\phi_d\sin\psi+\theta_d\cos\psi), \tag{8a}$$

$$\ddot{y}=\frac{-U_1}{m}(\theta_d\sin\psi-\phi_d\cos\psi), \tag{8b}$$

which after written in matrix form can be inverted to get

$$\begin{bmatrix}\phi_d\\ \theta_d\end{bmatrix}=\frac{m}{U_1}\begin{bmatrix}-\ddot{x}_d\sin\psi+\ddot{y}_d\cos\psi\\ -\ddot{x}_d\cos\psi-\ddot{y}_d\sin\psi\end{bmatrix}. \tag{9}$$

The standard continuous-time PID controller for position control in dimensions x and y is given by

$$\ddot{x}_d=k_P(x_d-x)+k_I\int(x_d-x)dt+k_D(\dot{x_d}-\dot{x}), \tag{10a}$$

$$\ddot{y}_d=k_P(y_d-y)+k_I\int(y_d-y)dt+k_D(\dot{y_d}-\dot{y}). \tag{10b}$$

Plugging the values of the desired accelerations $\ddot{x_d}$ and $\ddot{y_d}$ into (9), the desired roll and pitch angles ϕ_d and θ_d can be calculated. Input of each PID controller is the position error in axis x and y and the output is pitch and roll respectively. Then we send the desired values to the flight controller of the quadcopter. From the data obtained, considering the limitations of indoor space there are restrictions on the control signals.

Pitch and roll are in the range of [-0.1 ,0.1] so the maximum angles are 6 degrees. A saturation function is needed to ensure that the reference roll and pitch angles are within specified limits, i.e.,

$$\phi_d = sat(\phi),\ \theta_d = sat(\theta), \tag{11}$$

where

$$sat(v) = \begin{cases} v, & |v| \leq v_{max}, \\ sign(v)v_{max}, & |v| > v_{max}. \end{cases} \tag{12}$$

A. Discrete PID controller

The general form of continuous-time PID controllers is

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}, \tag{13}$$

which is used to control the angular rates and vertical position of the quadrotor. In our case, we use a discrete-time PID controller whose frequency depends on the UWB system update rate. The discrete-time version of the controllers is obtained by finite difference approximations:

$$\int_0^t e(\tau)d\tau \approx \sum_{i=1}^{k} e_i \Delta t, \tag{14}$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{\Delta t}, \tag{15}$$

where Δt is the sampling period (the UWB system update rate) and e_k the error at k_{th} sampling instant for $k = 1, 2, ...$ There are two forms of discrete PID controllers, the position form and the velocity form. The *position form* of the controller is given by

$$u_k = k_P e_k + k_I \Delta t \sum_{i=1}^{k} e_i + k_D \frac{(e_k - e_{k-1})}{\Delta t}$$

and the *velocity form* is [17]

$$u_k = u_{k-1} + k_P(e_k - e_{k-1}) + k_I e_k \Delta t + k_D \frac{(e_k - 2e_{k-1} + e_{k-2})}{\Delta t}.$$

The advantages of the velocity form over the position form are that it contains anti-reset windup because the summation of errors is not calculated. In our work, we tested the two forms and as we show in our results in the next section the position form is more precise. We tuned the PID controller manually starting from the proportional parameter. The output of the controller is angle in rad and is bounded between specified limits. We selected an initial proportional parameter so that the quadrotor does oscillate around the set-point and does not saturate too quickly because of the limitations.

V. Test and Results

A. Matlab Simulink model

To test the position controller we used a quadrotor Simulink model provided by MathWorks [18]. We set the position controller running at 20 Hz and the inner loop controllers running at 100 Hz. We added Gaussian noise with variance set to $0.00026cm^2$ to the position data, in order to emulate the UWB system, which we measured with the quadrotor on the ground. We set the desired point to be $(x, y) = (5, 2)$ and the initial position $(x, y) = (6, 1)$. The best PID parameters for the simulation after tuning are $k_P = 0.5$, $k_I = 0$ and $k_D = 1$. This results in a stable response without overshoot or oscillations, as it is shown in Fig. 4. The quadrotor is hovering in the desired position with no steady-state error; see Fig. 4.

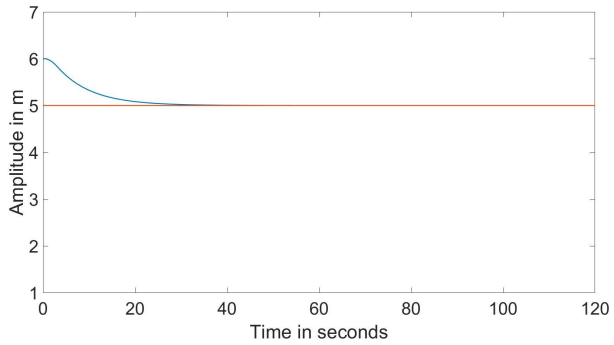

Fig. 4. Test without delay and noise. This is one of the fastest responses and the system is stable.

Then we add noise and delay to the measurement data to emulate the UWB system; see Fig. 5.

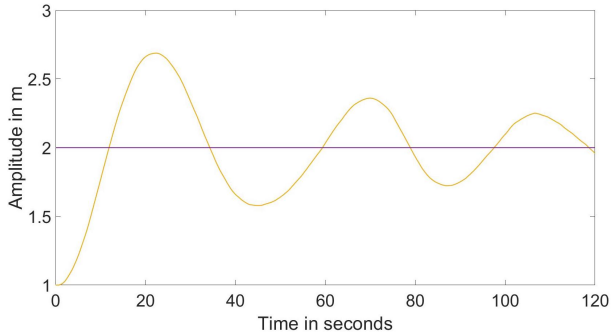

Fig. 5. Test with measurement noise and delay. These parameters result in slow decreasing oscillations.

The quadrotor is oscillating around desired position and the settling time is longer. Our conclusion is that the derivative part of the controller is important because it estimates the error at time proportional to the parameter k_D [19]. However it is sensitive to noise and we have to tune it carefully because of the UWB system measurement noise.

B. Flight tests

We used the ROS environment for the communication of the sensors, the onboard computer and the flight controller. The Pozyx tag communicates with the anchors, calculates its position. The ROS node reads the position and broadcasts it as a topic. The position controller node is subscribed to the topic and calculates the desired pitch and roll commands (we use steady yaw and altitude). Then the node sends the control message to the flight controller. The mean position calculation time is $28ms$, the mean publish time of the message is $2ms$ and the time of calculation of commands is $0.1ms$. The mean response time of the quadcopter when we send the desired

commands is $200-300ms$ (see Fig. 6). Our system was tested indoors with the fastest settings 64 bit preamble length and bitrate 6.81 Mbits/sec which result in a message update rate of 30Hz. The anchors are placed in certain locations in an area of $13.5m \times 5.2m$.

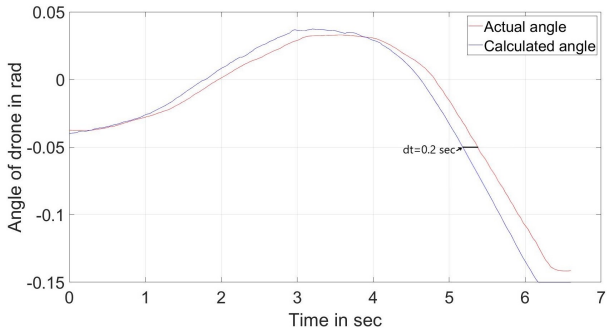

Fig. 6. Response time of the quadcopter.

C. Hover in steady position

1) Velocity form of PID controller: The desired position was set $x = 5m$ and $y = 2m$. The PID controller in velocity form is used in the first trials. The PID parameters are set to $k_P = 0.5$, $k_I = 0.01$ and $k_D = 4.5$ after manual tuning. The quadrotor hovers in an circle with radius $r = 0.5m$ around desired position; see Fig. 7.

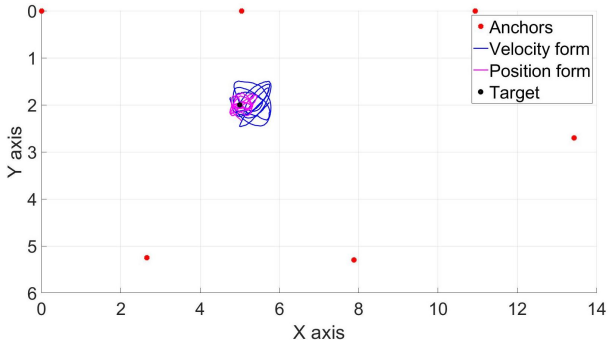

Fig. 7. Hover results using PID controller in velocity and position form.

2) Position form of PID controller: Next, we implemented the controller in position form. After manual tuning of our PID controller, we selected these parameters $k_P = 0.1$, $k_I = 0$ and $k_D = 4.5$. The quadrotor now hovers in an circle with radius $r = 15$cm around desired position (see Fig. 7). We use the position form of the PID Controller for our next tasks because of its better performance.

D. UWB Settings and system stability

In the next experiment, we change the UWB settings and observe how this impacts the system's performance. The signal transmission delay between the anchors and the tag is dependent to the system settings and the update rate of the positioning. A large delay may result in slow and oscillating response or even instability of the system. We operate the quadrotor in position hold mode where the quadcopter tries to keep a desired position using our controller. Then, we calculate the mean square error between the desired and actual position for different UWB settings; the outcome is depicted in Fig. 8.

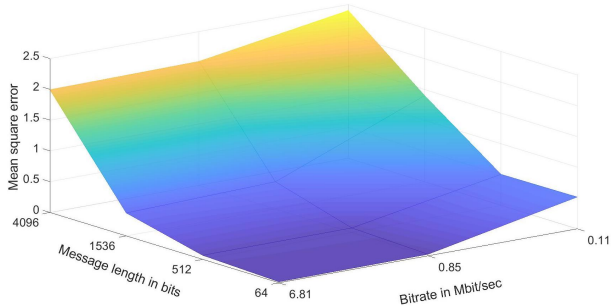

Fig. 8. Mean square error for different message length and bitrate.

In Fig. 9, we show the quadrotor trajectory for the fastest and slowest message passing of the available UWB settings.

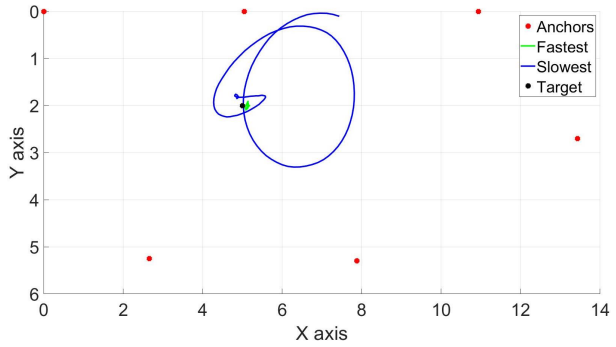

Fig. 9. Trajectories of the quadcopter using the slowest and fastest message passing of the available UWB settings. The slowest settings are with message length = 4096 bits and bitrate = 0.11 Mbits/sec and the fastest are with message length = 64 bits and bitrate = 6.81 Mbits/sec.

As expected, the shorter the message length and the faster the bitrate, the smaller error. As a result, the quadrotor hovers close the desired position. For message length longer than 1536 bits the error is very large and the trajectory of the quadrotor is oscillatory with increasing amplitude, i.e., the system becomes unstable.

E. Line trajectory

The next task is to follow certain desired trajectories and find the error between the actual and desired trajectory. The first trajectory is to follow a straight line trajectory.

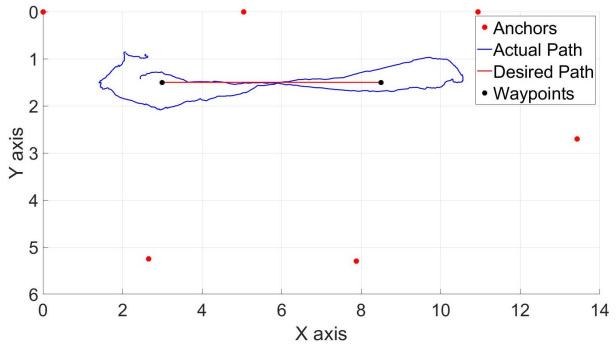

Fig. 10. Straight line trajectory.

As we see in Fig. 10, the quadcopter follows a longer path instead of the desired line because of the communication delay and the angle limits that do not allow the quadcopter to change direction faster when it is in the end of the line. Our next task is to make the quadcopter follow an orthogonal trajectory as we see in Fig. 11.

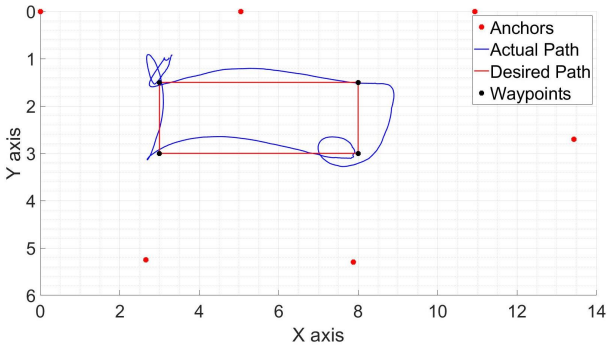

Fig. 11. Square trajectory.

The angle limitations and the delayed response do not allow the quadrotor to follow the path with high accuracy. Hence, the quadrotor in this scenario does not follow the desired trajectory with high precision.

VI. Conclusions and Future Directions

In this paper, we proposed a PID control design for indoor positioning and navigation of a quadrotor by means of a wireless positioning system using UWB measurements. The controller designed investigates the trade off between latency and reliability. The performance is experimentally investigated in a hardware testbed. The effectiveness of the position controller is verified and illustrated through different flight tasks: In the first task, in which the quadrotor had to hover at a desired position, it was shown that the PID controller had a good performance (quadrotor remained in a radius of under $15cm$). In the second task, in which the quadrotor had to follow a specific trajectory, the performance of the PID controller was not as accurate as one would expect, given the first task. This is mainly due to the fact that in the hovering task, the system works in its linear regime and a PID controller is adequate, whereas in the second task the quadrotor possibly works in a nonlinear regime.

The fact that the quadrotor was unable to follow simple trajectories accurately suggests that estimation techniques and more advanced controllers should be deployed. For the system to operate outside its linear regime, nonlinear [15] or adaptive controllers [20] might be necessary. The performance degradation was partly because of UWB message delays, which remain a challenge for accurate positioning.

Furthermore, part of our future work is to consider collaborative localization algorithms [21] and integrate the UWB with the IMU measurements in order to achieve higher precision.

References

[1] A. Nemra and N. Aouf, "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering," *IEEE Sensors J.*, vol. 10, no. 4, pp. 789–798, 2010.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-UAV testbed," *IEEE Robot. Automat. Mag.*, vol. 17, no. 3, pp. 56–65, 2010.

[3] A. Garcia, E. Mattison, and K. Ghose, "High-speed vision-based autonomous indoor navigation of a quadcopter," in *Int. Conf. on Unmanned Aircraft Syst.*, 2015, pp. 338–347.

[4] C. Wang, K. Li, G. Liang, H. Chen, S. Huang, and X. Wu, "A heterogeneous sensing system-based method for unmanned aerial vehicle indoor positioning," *Sensors*, vol. 17, no. 8, p. 1842, 2017.

[5] F. Yu, G. Chen, N. Fan, Y. Song, and L. Zhu, "Autonomous flight control law for an indoor UAV quadrotor," in *Chinese Control and Decision Conf.*, 2017, pp. 6767–6771.

[6] K. Cisek, A. Zolich, K. Klausen, and T. A. Johansen, "Ultra-wide band Real time Location Systems: Practical implementation and UAV performance evaluation," in *Workshop on Res., Edu. and Develop. of Unmanned Aerial Syst.*, 2017, pp. 204–209.

[7] K. Guo, Z. Qiu, C. Miao, A. H. Zaini, C.-L. Chen, W. Meng, and L. Xie, "Ultra-wideband-based localization for quadcopter navigation," *Unmanned Systems*, vol. 4, no. 01, pp. 23–34, 2016.

[8] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *IEEE Int. Conf. on Robot. and Automation*, 2015, pp. 1730–1736.

[9] S. Fahandezh-Saadi and M. W. Mueller, "An algorithm for real-time restructuring of a ranging-based localization network," *arXiv preprint arXiv:1804.09833*, 2018.

[10] B. Dewberry and M. Einhorn, "Indoor aerial vehicle navigation using ultra wideband active two-way ranging," in *AUVSI Xponential Conf.*, 2016.

[11] J. Tiemann, F. Schweikowski, and C. Wietfeld, "Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments," in *IEEE Int.l Conf. on Indoor Positioning and Indoor Navigation*, 2015, pp. 1–7.

[12] A. Benini, A. Mancini, and S. Longhi, "An IMU/UWB/vision-based extended Kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network," *J. of Intell. & Robot. Syst.*, vol. 70, no. 1-4, pp. 461–476, 2013.

[13] D. E. Manolakis, "Efficient solution and performance analysis of 3-D position estimation by trilateration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1239–1248, Oct 1996.

[14] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *J. of Intell. & Robot. Syst.*, vol. 70, no. 1-4, pp. 1–12, 2013.

[15] H. talla Mohamed Nabil Elkholy, "Dynamic modeling and control of a quadrotor using linear and nonlinear approaches," Master's thesis, School of Engineering Interdisciplinary Program, The American University in Cairo, Egypt, 2013.

[16] H. Hou, J. Zhuang, H. Xia, G. Wang, and D. Yu, "A simple controller of minimize quad-rotor vehicle," in *IEEE Int. Conf. on Mechatronics and Automation*, Aug 2010, pp. 1701–1706.

[17] H. S. Khan and M. B. Kadri, "Position control of quadrotor by embedded PID control with hardware in loop simulation," in *IEEE Int. Multi-Topic Conf.*, 2014, pp. 395–400.

[18] U. Pattacini, "Displacement control of a quadcopter simulink model." [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/51076-displacement-control-of-a-quadcopter

[19] T. Hägglund, "A dead-time compensating three-term controller," *IFAC Proc. Volumes*, vol. 24, no. 3, pp. 1167–1172, 1991.

[20] A. L'Afflitto, R. B. Anderson, and K. Mohammadi, "An Introduction to Nonlinear Robust Control for Unmanned Quadrotor Aircraft: How to Design Control Algorithms for Quadrotors Using Sliding Mode Control and Adaptive Control Techniques [Focus on Education]," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 102–121, June 2018.

[21] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative Sensor Network Localization: Algorithms and Practical Issues," *Proceedings of the IEEE*, vol. 106, no. 6, pp. 1089–1114, June 2018.