Dynamic Target Tracking and Obstacle Avoidance using a Drone

Conference Paper · December 2015

DOI: 10.1007/978-3-319-27857-5_76

2 authors:

Alexander C. Woods
University of Nevada, Reno

Hung Manh La
University of Nevada, Reno

The 11th International Symposium on Visual Computing (ISVC),
December 14-16, 2015 Las Vegas, Nevada, USA.

Dynamic Target Tracking and Obstacle Avoidance using a Drone

Alexander C. Woods and Hung M. La(✉)

Advanced Robotics and Automation Laboratory,
Department of Computer Science and Engineering,
University of Nevada, Reno, NV 89557, USA
hla@unr.edu

Abstract. This paper focuses on tracking dynamic targets using a low cost, commercially available drone. The approach presented utilizes a computationally simple potential field controller expanded to operate not only on relative positions, but also relative velocities. A brief background on potential field methods is given, and the design and implementation of the proposed controller is presented. Experimental results using an external motion capture system for localization demonstrate the ability of the drone to track a dynamic target in real time as well as avoid obstacles in its way.

1 Introduction

This paper focuses on dynamic target tracking and obstacle avoidance on a quadrotor drone. Drones are useful in many practical applications such as disaster relief efforts, infrastructure inspections, search and rescue, mapping, and military or law enforcement surveillance. However, in order to perform any of these tasks, a drone must be able to perform both localization as well as effective navigation and obstacle avoidance. The contribution of this paper is the development of a drone control system which allows the drone to track a dynamic target while simultaneously avoiding obstacles using an expanded potential field control method. A demonstration of the capabilities is performed utilizing the ARDrone 2.0 shown in Fig. 1 and an external motion capture system [1].

Currently, two main challenges prevent drones from being used regularly for application such as those mentioned. The first is localization. GPS is a great tool for localization in environments which allow communication with satellites, and there has been a lot of research activity in the field of localization in GPS-denied environments. For example, light detection and ranging (LIDAR) is a promising option [2–4] which has been used in simultaneous localization and mapping (SLAM) [5]. Another method is computer vision, utilizing both forward and downward looking stereo-cameras [6]. Additionally, tools such as Microsoft's Kinect sensor have been used which provide depth information much like LIDAR, but with a much larger sensing area [7]. All of these options, and more, provide drones with location information about themselves and their environment, including obstacles.

G. Bebis et al. (Eds.): ISVC 2015, Part I, LNCS 9474, pp. 857–866, 2015.
DOI: 10.1007/978-3-319-27857-5_76

Fig. 1. A low cost, commercially available quadrotor drone is used as the experimental platform for testing and demonstrating the effectiveness of the proposed control method.

The second challenge is accurate navigation and crash avoidance which enable the drone to perform any given task safely and efficiently. Similar to localization, this is also a field which has attracted much research. Many use multi-drone flocking strategies in order to track and maintain contact with a target for information gathering purposes [8,9]. Several methods have been developed for fixed-wing aircrafts, utilizing numerous methods ranging from Lyapunov-LaSalle to seventh order Bézier curves [10–12]. For use on quadrotor drones, a PID controller has been used for position control in a hover state coupled with a three dimensional PD controller for trajectory tracking [13,14] with good results. In this paper, a novel potential field controller is presented which has been expanded to include not only relative position but also relative velocity in order to meet the unique challenges presented by drones. By expanding the method to include more than just two dimensional position data, greater performance can be achieved and allow the drone to perform in fast moving environments.

This paper is organized as follows. Section 2 describes the problem definition, assumptions, constrains and evaluation criteria for the paper. Section 3 provides a brief background on potential field methods, and their limitations. Section 4 discusses the design of the controller used in this paper. Section 5 presents the experimental results of the paper, and an evaluation of the performance of the controller. Finally, Sect. 6 provides a brief conclusion, with recommendations for future work.

2 Problem Definition

The goals of this paper are to develop a control system for a drone which allows it to safely navigate to, and follow a dynamic human target. The drone should never run into the target or any obstacles in its vicinity. For the purposes of this paper, we assume that the position of the drone and the target are known. In this case, we utilize an external motion tracking system [1] to provide position feedback, which is this paper's main constraint. It is important to note that the approach presented maybe be adapted to work with onboard sensing in future work.

To evaluate the success or failure of this paper, the drone should be able to navigate successfully from a starting location to a target location, while avoiding obstacles along the way. Additionally, the drone should be able to track the target while it moves around the testing environment.

3 Potential Field Navigation

Because of their simplicity and elegance, potential field methods are often used for navigation of ground robots [15–17]. Potential fields are aptly named, because they use attractive and repulsive potential equations to draw the drone toward a goal (attractive potential) or push it away from an obstacle (repulsive potential). For example, imagine a stretched spring which connects a drone and a target. Naturally, the spring will draw the drone to the target location.

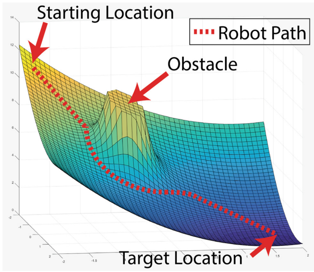

Fig. 2. An example of a traditional potential field which can be used for navigating toward a target while avoiding obstacles.

Conveniently, potential fields for both attractive and repulsive forces can be summed together, to produce a field such as that shown in Fig. 2. The example shown includes only one repulsive field, but multiple fields representing multiple obstacles could easily be added as they are sensed.

Traditionally, potential forces work in the x and y spatial dimensions, and are defined by a quadratic function given by

$$U_{att}(q_d) = \frac{1}{2}\lambda_1 d^2(q_d, q_t), \tag{1}$$

where λ_1 is positive scale factor, and $d(q_d, q_t)$ is the relative distance between the drone and the target. In order to achieve the target location in this scenario, the potential function should be minimized, thus the velocity of the drone should be

$$\begin{aligned} p_d^{att}(q_d) &= \nabla U_{att}(q_d) = \nabla\Big(\frac{1}{2}\lambda_1 d^2(q_d, q_t)\Big), \\ &= \frac{1}{2}\lambda_1 \nabla d^2(q_d, q_t), \\ &= \lambda_1(q_t - q_d). \end{aligned} \tag{2}$$

The velocity in (2) is appropriate for a stationary target but for a moving target, the velocity of the target should be also taken into account. If we consider that the drone should move with the same velocity as the target, in addition to the motion dictated by the attractive potential, then (2) becomes

$$p_d^{att}(q_d) = \lambda_1(q_t - q_d) + p_t, \tag{3}$$

where p_t is the velocity of the target. Through simulation and mathematical proofs not presented here, it can be shown that the relative position between the drone and target converges to zero using the method described. Thus, this method forms the foundation for the controller developed in this paper.

To account for obstacles, a potential function is generated which grows increasingly repulsive as the distance to the obstacle decreases to zero. Thus, the repulsive potential function is given by

$$U_{rep}(q_d) = \frac{1}{2}\eta_1 \frac{1}{d^2(q_d, q_o)}, \tag{4}$$

for which the resulting velocity of the drone is in the direction of the gradient, and is given by

$$p_d^{rep}(q_d) = \nabla U_{rep}(q_d) = -\eta_1 \frac{1}{(q_o - q_d)^3}, \tag{5}$$

where η_1 is again a scaling factor, and $d(q_d, q_o)$ is the relative distance between the drone and the obstacle. Because the drone should never actually come in contact with the obstacle, the relative distance should be computed between the drone and the a minimum distance it can be from the obstacle.

4 Controller Design

Because the potential field methods presented above are developed for ground robots, they do not address many of the factors that must be accounted for when designing a controller for aerial systems. For example, drones move very quickly and are inherently unstable which means they cannot simply move to a particular location and stop moving. They are consistently making fine adjustments to their position and velocity.

In order to account for factors unique to aerial platforms, not only is the relative position between the drone and the target taken into account, but also the relative velocity between the two. Using this new information, a potential function based on the relative velocity is given by

$$U_{att}(p_d) = \frac{1}{2}\lambda_2 v^2(p_d, p_t), \tag{6}$$

where λ_2 is a positive scale factor, and $v(p_d, p_t)$ is the relative velocity between the drone and the target. Again, this function should be minimized by moving in the direction of the gradient. Thus the speed of the drone due to attractive potential from the target should be the sum of the gradient of this potential function and (3) which yields

$$p_d^{att}(q_d, p_d) = \lambda_1(q_t - q_d) + p_t + \lambda_2(p_d - p_t), \tag{7}$$

Similarly, the potential function for obstacle avoidance derived in Sect. 3 should be augmented to include the a potential function for relative velocity between the drone and the obstacle. This potential function should react when the drone is moving toward the obstacle, but have no effect if the drone is moving away. Thus a piecewise potential function is generated and is described by

$$U_{rep}(p_d) = \begin{cases} -\frac{1}{2}\eta_2 v^2(p_d, p_o) & : (p_o - p_d) < 0 \\ 0 & : (p_o - p_d) \geq 0 \end{cases} \tag{8}$$

and the resulting total velocity from repulsive forces is

$$p_d^{rep}(q_d, p_d) = -\eta_1 \frac{1}{(q_o - q_d)^3} - \eta_2(p_o - p_d). \tag{9}$$

Now that we have the velocity contributions from all of the attractive and repulsive potentials, we simply add them together to get the final velocity of the drone.

$$p_d(q_d, p_d) = p_d^{att} + p_d^{rep} \tag{10}$$

This velocity is calculated in the world frame, so must be transformed to the body frame of the drone. Given the heading of the drone, θ_d, the speed in the x_{body} and y_{body} is calculated using

$$\begin{aligned} v_{x,body} &= p_d cos(\theta_d) + p_d sin(\theta_d) \\ v_{y,body} &= -p_d sin(\theta_d) + p_d cos(\theta_d) \end{aligned} \tag{11}$$

This controller will seek out a moving target, and will also avoid obstacles that are in close proximity.

5 Experimental Results

The experimental platform used for this paper was the commercially available Parrot ARDrone 2.0 shown in Fig. 1 which has built in wifi connectivity which allows an easy way to control it through a computer interface, such as Matlab. In addition to the ARDrone, an external motion capture system was used to provide localization capabilities for the drone, target, and obstacle. The lab environment with drone, obstacle, target, and external tracking system is shown in Fig. 3.

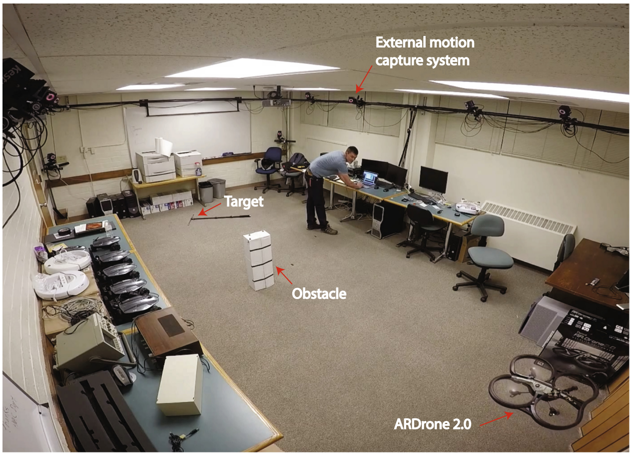

Fig. 3. The experimental environment with a drone, target, obstacle, and external motion tracking system is shown.

In order to control the drone and display its location along with the target and any obstacles present, the Matlab GUI shown in Fig. 4 was created. It allows the user to determine when the drone takes off, lands, or tracks the target. This GUI is critical in efficient testing of the drone. Additionally, for the safety of the drone, if the controller does not behave as expected the user can request that the drone simply hover in place to avoid fly-aways.

In order to test the performance of the controller, several experiments were performed. First, the ARDrone is placed across the room from a stationary target, and is commanded to take off and reach the goal location. The performance of the controller here is evaluated by the overshoot and settling time of the system. Second, the first experiment is repeated but with an obstacle introduced. The performance of the controller in this case is evaluated by the success or failure of reaching the target without running into the obstacle. Finally, the ARDrone is commanded to follow the target as it moves about in the lab environment. This task is harder to evaluate, and a demonstration of the drone performing the requested action is used to verify that this functionality works. Future works should address how to evaluate the performance of this method quantitatively.

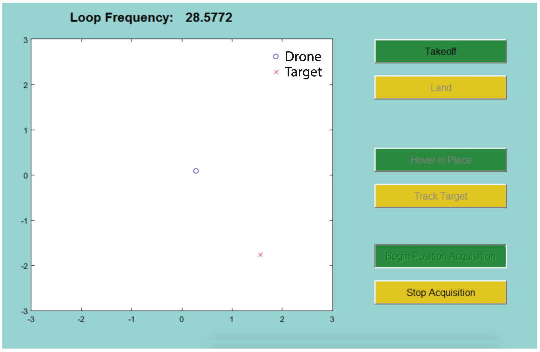

Fig. 4. A Matlab GUI was created to show the positions of the drone, target, and any obstacles present. It also allows the user to control when the drone takes off, lands, tracks the target, or simply hovers in place.

During the first experiment, the drone was placed approximately 4.2 meters from the target's location. The drone was requested to fly to 1 m away from the target location, since the purpose of this is to eventually track a human. The drone's response shown in Fig. 5 demonstrates the capability of the drone to achieve a goal position effectively. Starting at approximately 7.75 seconds, the drone enters an autonomous mode, and achieves stable hover 1 m away from the target at approximately 12.75 seconds. It is important to note that while the drone did overshoot it's goal location, it did not overshoot enough to get close to hitting the target. The closest that the drone got to the target was just under 0.75 meters.

In the second experiment, the drone was placed approximately 5.2 meters away from the target, and an obstacle was located in the path lying directly between the drone and the target. Similar to the first test, the drone's mission was to fly to within 1 m of the target, this time while avoiding the obstacle and still achieving the task. As the drone begins moving towards the target, it also moves towards the obstacle. Because of the repulsive forces generated by the

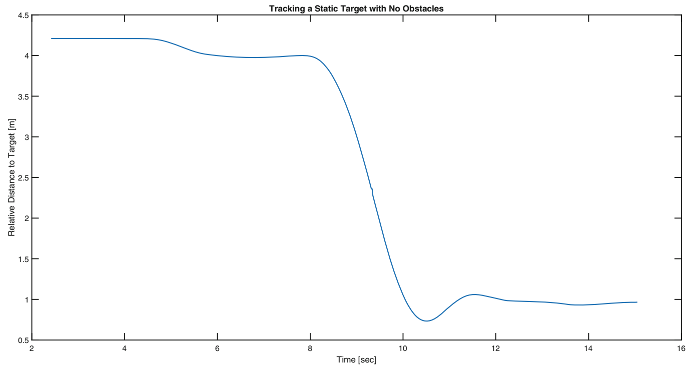

Fig. 5. The results of tracking a static target with no obstacles are very good. With an initial condition of approximately 3.2 m, the drone achieves position in under 5 seconds.

drones relative location and velocity with respect to the obstacle, the drone is elegantly "pushed" around the obstacle and still makes it to the target location. Figure 6 shows the results of this test, demonstrating that the drone maintains a safe distance from the obstacle (1 meter minimum) and also achieves the goal.

In the third experiment the drone is directed to follow the target around, while maintaining a safe distance at all times. In addition, the drone should always face the target. This task was performed several times to evaluate the performance. In each of the tests the drone successfully completes the task. Even under extreme circumstances (e.g., very fast maneuvers) the drone is able to recover and maintain the desired behavior. Figure 7 shows frames from a video [18] taken of the drone performing this task. In the video it can clearly

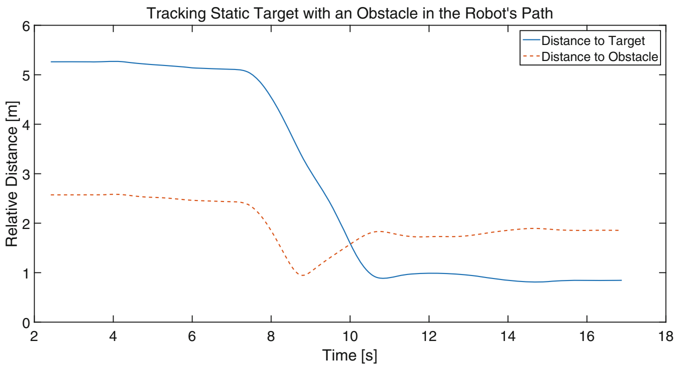

Fig. 6. The results of tracking a static target with an obstacle in the way demonstrates the controllers effectiveness at avoiding collisions.

Fig. 7. The third experiment demonstrates the drone tracking a dynamic target, while maintaining the proper heading to always face the target.

be seen that the drone follows the target around while always maintaining the proper heading to face the target.

In addition to tracking a dynamic target moving in an arbitrary pattern, the drone was instructed to follow the target as it moved in an approximate rectangle around the lab. The results shown in Fig. 8 illustrate the path of the drone as it follows the target through the pattern. As shown, the drone does in fact track the rectangle as instructed, with slight deviations. Because the path of the target was moved manually by a person as in previous experiments, the target trajectory is not a perfect rectangle either. In future work, a set trajectory will be established without human error which would allow repeatability to be addressed as well as a much better analysis of the tracking error.

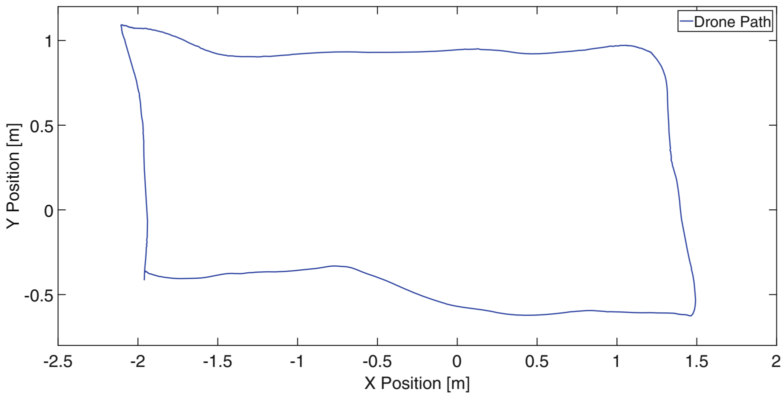

Fig. 8. The final experiment has the drone track a target which moves in an approximate rectangle. As shown, the drone does track, but because the desired trajectory is human-controlled the reference is not perfect. Future work will include using a predetermined and set trajectory.

6 Conclusion

This paper presented a novel, extended potential field controller designed to enable a drone to track a moving target in a dynamic environment. An experiment setup comprising of an ARDrone 2.0 and an external motion tracking system was used to implement the controller and evaluate the performance. The drone was able to successfully track stationary targets, with and without obstacles in it's path, as well as dynamic targets moving about in the lab.

Future work to build upon this paper will include determining a quantifiable evaluation method for the performance of dynamic target tracking. More importantly, because this system relies upon an external tracking system future work should include a method for estimating the state of both the drone and the target using onboard sensing and systems. Potential solutions might be LIDAR, optic flow sensing, and more.

Acknowledgements. The authors would like to specially thank the Motion Analysis Corporation for their support of the Motion Tracking System (MTS) setup at the Advanced Robotics and Automation (ARA) Lab at the University of Nevada, Reno. This project is partially supported by University of Nevada, Reno and NSF-NRI grant number 1426828.

References

1. Motion analysis systems. http://www.motionanalysis.com
2. Shaohua, M., Jinwu, X., Zhangping, L.: Navigation of micro aerial vehicle in unknown environments. In: 2013 25th Chinese Control and Decision Conference (CCDC), pp. 322–327 (2013)

3. Sa, I., Corke, P.: System identification, estimation and control for a cost effective open-source quadcopter. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 2202–2209, May 2012
4. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained mav. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 20–25, May 2011
5. Grzonka, S., Grisetti, G., Burgard, W.: A fully autonomous indoor quadrotor. IEEE Trans. Robot. **28**(1), 90–100 (2012)
6. Meier, L., Tanskanen, P., Heng, L., Lee, G., Fraundorfer, F., Pollefeys, M.: Pixhawk: a micro aerial vehicle design for autonomous flight using onboard computer vision. Auton. Robot. **33**(1–2), 21–39 (2012)
7. Shen, S., Michael, N., Kumar, V.: Autonomous indoor 3d exploration with a micro-aerial vehicle. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 9–15 (2012)
8. Shanmugavel, M., Tsourdos, A., White, B., Zbikowski, R.: Co-operative path planning of multiple uavs using dubins paths with clothoid arcs. Control Eng. Pract. **18**(9), 1084–1092 (2010)
9. Wallar, A., Plaku, E., Sofge, D.: Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas. IEEE Trans. Autom. Sci. Eng. **12**(3), 969–980 (2015)
10. Neto, A.A., Macharet, D., Campos, M.: Feasible path planning for fixed-wing uavs using seventh order bezier curves. J. Braz. Comput. Soc. **19**(2), 193–203 (2013)
11. Altmann, A., Niendorf, M., Bednar, M., Reichel, R.: Improved 3d interpolation-based path planning for a fixed-wing unmanned aircraft. J. Intell. Robot. Syst. **76**(1), 185–197 (2014)
12. Dogancay, K.: Uav path planning for passive emitter localization. IEEE Trans. Aerosp. Electron. Syst. **48**(2), 1150–1166 (2012)
13. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The grasp multiple micro-uav testbed. IEEE Robot. Autom. Mag. **17**(3), 56–65 (2010)
14. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2520–2525, May 2011
15. La, H.M., Lim, R.S., Sheng, W., Chen, J.: Cooperative flocking and learning in multi-robot systems for predator avoidance. In: 2013 IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER), pp. 337–342, May 2013
16. La, H.M., Sheng, W.: Multi-agent motion control in cluttered and noisy environments. J. Commun. **8**(1), 32 (2013)
17. La, H.M., Sheng, W.: Dynamic target tracking and observing in a mobile sensor network. Robot. Auton. Syst. **60**(7), 996 (2012)
18. Woods, A.C.: Dynamic target tracking with ardrone. https://youtu.be/v85hs8-uc1s