A Novel Extended Potential Field Controller for Use on Aerial Robots

Alexander C. Woods, Hung M. La[†], and Quang P. Ha

***Abstract*—Unmanned Aerial Vehicles (UAV), commonly known as drones, have many potential uses in real world applications. Drones require advanced planning and navigation algorithms to enable them to safely move through and interact with the world around them. This paper presents an extended potential field controller (ePFC) which enables an aerial robot, or drone, to safely track a dynamic target location while simultaneously avoiding any obstacles in its path. The ePFC outperforms a traditional potential field controller (PFC) with smoother tracking paths and shorter settling times. The proposed ePFC's stability is evaluated by Lyapunov approach, and its performance is simulated in a Matlab environment. Finally, the controller is implemented on an experimental platform in a laboratory environment which demonstrates the effectiveness of the controller.**

I. Introduction

THIS paper focuses on dynamic target tracking and obstacle avoidance on a quadrotor drone such as the one shown in Fig. 1. Recent advances in the field of unmanned autonomous systems (UAS) have drastically increased the potential uses of both unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV). UAS can be utilized in situations which may be hazardous to human operators in ground vehicles or pilots in traditional aircraft, such as assisting wild land fire fighters [1]–[5], search and rescue operations in unsafe conditions or locations [6]–[10], and disaster relief efforts [11]–[13]. Additionally, UAS can be used in repetitive or tedious work where a human operator may loose focus such as infrastructure inspection [14]–[16], agricultural inspections [17], [18], and environmental sensing [19].

Although the field of UAS has grown rapidly, it is still hindered by many problems which limit their use in real world applications. The challenge of localizing in GPS-denied environments has been approached by a multitude of research groups across the world, and there are several methods which have been to address this. One of several promising on-board sensing methods is light detection and ranging (LIDAR). One group employed a reflexive algorithm in combination with a LIDAR sensor for simulating navigation through an unknown environment [20]. Another group developed a multilevel simultaneous localization and mapping (SLAM) algorithm which utilized LIDAR as its primary sensing method [21]. Other groups used LIDAR on autonomous vehicles for control and multi-floor navigation [22], [23].

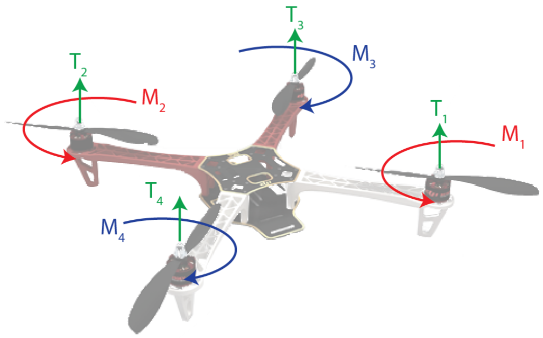

Fig. 1. A quadrotor drone achieves lift by utilizing four rotors which each produce a coupled moment about the axis of rotation. Because they only have four actuators, quadrotors are inherently under-actuated. This makes control a challenge, but also means they can perform very agile maneuvers which gives them an advantage over fixed wing platforms in confined environments.

Another major area of research for localization in GPS-denied environments is computer vision. One group used a single camera, looking at an object of known size to determine the drone's location [24]. Another group utilized a combination of LIDAR and a Microsoft Kinect sensor to explore an unknown environment [25]. Several other groups successfully used unique variations of computer vision methods as a means of localization [26], [27], and it is proving to be a very promising method of operating in GPS-denied environments.

In addition to advanced sensing capabilities, UAS also require planning and navigation algorithms to safely move through and interact with the world around them. Trajectory generation for aerial robots has been accomplished through methods such as minimizing snap, the second derivative of acceleration [28], [29]. Given keyframes consisting of a position in space coupled with a yaw angle, this method is able to generate very smooth, optimal trajectories. Other groups successfully applied methods utilizing Voronoi diagrams [24], [30], receding horizons in relatively unrestricted environments [31], high order parametric curves [32], and 3D interpolation [33].

However, many platforms do not have the luxury of a very powerful processor and solving complex algorithms cannot practically be performed by an offboard computer. Therefore, the contribution of this paper is to propose an ePFC as a navigation method which is computationally inexpensive, can react quickly to the environment, and which can be deployed onboard any platform with adequate sensing capabilities. The developed controller's stability is evaluated, and its performance is both simulated and demonstrated experimentally.

The remainder of this paper is organized as follows. Section II provides a brief background on potential field methods, discusses the design of the controller used in this paper,

[†]Corresponding author; E-mail: hla@unr.edu.

A. Woods and H. La are with the Advanced Robotics and Automation Laboratory, Department of Computer Science and Engineering, University of Nevada - Reno, Reno, NV 89557-0312 USA

Q. Ha is with the School of Electrical, Mechanical and Mechatronic Systems, University of Technology Sydney, Australia. Email: Quang.Ha@uts.edu.au

and demonstrates the stability of the system using a Lyapunov approach. Section III presents simulation results of the controller implemented in a Matlab environment. Section IV discusses the experimental quadrotor platform and the testing environment. Section V presents experimental results, and an evaluation of the controller's performance. Finally, section VI provides a brief conclusion, with recommendations for future work.

II. Controller Design and Stability

Because of their simplicity and elegance, potential field controllers (PFCs) are often used for navigation of ground robots [34]–[36]. 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.

Conveniently, potential fields for both attractive and repulsive forces can be summed together, to produce a field such as the one shown in Fig. 2. This figure illustrates how a robot can navigate toward a target location while simultaneously avoiding obstacles in its path.

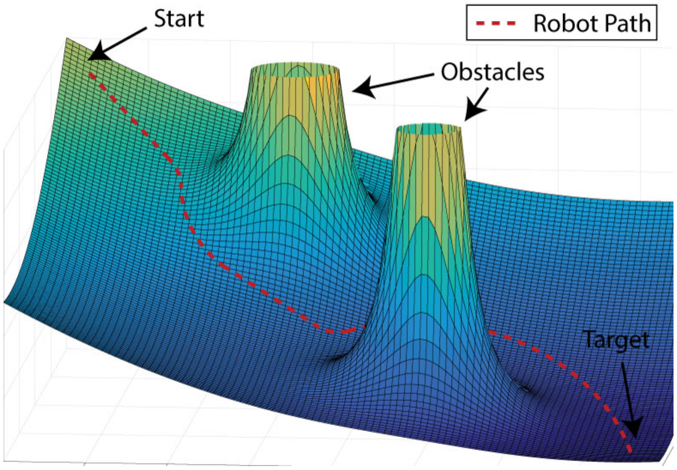

Fig. 2. An example of a traditional potential field which can be used for navigating toward a target while avoiding multiple obstacles.

Let us denote $p_d = [x_d, y_d, z_d]^T$ and $p_t = [x_t, y_t, z_t]^T$ as the position vector of drone and target, respectively. The relative distance vector between the drone and the target is then

$$\begin{aligned} p_{dt} &= [x_{dt}, y_{dt}, z_{dt}]^T, \\ &= [x_d, y_d, z_d]^T - [x_t, y_t, z_t]^T. \end{aligned} \tag{1}$$

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

$$U_{att}(p_d, p_t) = \frac{1}{2}\lambda_1 \|p_{dt}\|^2, \tag{2}$$

where λ_1 is positive scale factor, and $\|p_{dt}\|$ is the magnitude of the relative distance between the drone and the target, which is given by

$$\|p_{dt}\| = \sqrt{(x_{dt})^2 + (y_{dt})^2 + (z_{dt})^2}. \tag{3}$$

As shown in Fig. 2, the target location is always a minimum of the overall potential field. Therefore, in order to achieve the target location, the UAS should always move "downhill." The direction and magnitude of the desired movement can be computed by finding the negative gradient of the potential field, given by

$$\begin{aligned} v_d^{att}(p_d, p_t) &= -\nabla U_{att}(p_{dt}), \\ &= -\lambda_1(p_d - p_t), \end{aligned} \tag{4}$$

where v_d^{att} is the desired velocity due to the attractive position potential.

This is the classic form of a simple attractive potential field controller. However, this does not yet take into account obstacles or other sources of repulsive potential. The repulsive potential is proportional to the inverse square of the distance between the drone and the obstacle and is given by

$$U_{rep}(p_d, p_o) = \frac{1}{2}\eta_1 \frac{1}{\|p_{do}\|^2}, \tag{5}$$

where η_1 is positive scale factor, and $\|p_{do}\|$ is the magnitude of the relative distance between the drone and the obstacle.

To find the desired velocity, we again take the gradient of the potential field which yields

$$\begin{aligned} v_d^{rep}(p_d, p_o) &= \nabla U_{rep}(p_d, p_o), \\ &= \eta_1 \frac{p_d - p_o}{\|p_{do}\|^3}, \end{aligned} \tag{6}$$

where v_d^{rep} is the desired velocity due to the repulsive position potential. A complete traditional potential field controller is the sum of (4) and (6) which yields

$$v_d^{PFC}(p_d, p_t, p_o) = -\lambda_1(p_t - p_d) + \sum_{i=0}^{n} \eta_1 \frac{p_d^i - p_o^i}{\|p_{do}^i\|^3}, \tag{7}$$

where n is the number of obstacles present in the environment.

This controller enables a ground robot to track stationary or dynamic targets, while avoiding any obstacles in its path. However, when applied to an agile, aerial system such as a quadrotor, the controller's performance is quite poor as shown in simulations presented in Section III.

To adapt potential field methods for use on an aerial robot, this paper presents an extended potential field controller (ePFC) which utilizes the same concepts found in a traditional PFC, but applied to relative velocity rather than position. If we consider that we are tracking a dynamic target, then the desired velocity will be that of the target. In this case, the attractive potential will be defined as the quadratic function given by

$$U_{att}(v_d, v_t) = \frac{1}{2}\lambda_2 \|v_{dt}\|^2, \tag{8}$$

where λ_2 is positive scale factor, and $\|v_{dt}\|$ is the magnitude of the relative velocity between the drone velocity, v_d, and the target velocity, v_d, which is given by

$$\|v_{dt}\| = \sqrt{(\dot{x}_{dt})^2 + (\dot{y}_{dt})^2 + (\dot{z}_{dt})^2}. \tag{9}$$

As in the traditional potential field controller, we wish to minimize the relative velocity potential thus resulting in a matched velocity between the drone and the target. Similar to the traditional controller, we find the desired velocity of the drone by calculating the negative gradient, which we find to be

$$\begin{aligned} v_d^{att}(v_d, v_t) &= -\nabla U_{att}(v_d, v_t), \\ &= -\frac{\partial U_{att}}{\partial \dot{x}} - \frac{\partial U_{att}}{\partial \dot{y}} - \frac{\partial U_{att}}{\partial \dot{z}}, \\ &= -\nabla\Big(\frac{1}{2}\lambda_2 \|v_{dt}\|^2)\Big), \\ &= -\frac{1}{2}\lambda_2 \nabla \|v_{dt}\|^2, \\ &= -\frac{1}{2}\lambda_2 \nabla(\dot{x}_{dt}^2 + \dot{y}_{dt}^2 + \dot{z}_{dt}^2), \\ &= -\lambda_2(\dot{x}_{dt} + \dot{y}_{dt} + \dot{z}_{dt}), \\ &= -\lambda_2(v_d - v_t). \end{aligned} \tag{10}$$

The repulsive velocity potential between the drone and an obstacle is given by

$$U_{rep}(v_d, v_o) = \frac{1}{2}\eta_2 \frac{1}{\|v_{do}\|^2}, \tag{11}$$

where η_2 is positive scale factor, and $\|v_{do}\|^2$ is again the magnitude of the relative velocity between the drone velocity, v_d, and the obstacle velocity, v_o. The corresponding velocity for this potential function is found by

$$\begin{aligned} v_d^{rep}(v_d, v_o) &= \nabla U_{rep}(v_d, v_o), \\ &= \frac{\partial U_{rep}}{\partial \dot{x}} + \frac{\partial U_{rep}}{\partial \dot{y}} + \frac{\partial U_{rep}}{\partial \dot{z}}, \\ &= -\nabla\Big(\frac{1}{2}\eta_2 \frac{1}{\|v_{do}\|^2}\Big), \\ &= -\frac{1}{2}\eta_2 \nabla \frac{1}{\|v_{do}\|^2}, \\ &= -\frac{1}{2}\eta_2 \nabla \frac{1}{\dot{x}_{do}^2 + \dot{y}_{do}^2 + \dot{z}_{do}^2}, \\ &= \eta_2 \frac{v_d - v_o}{(\dot{x}_{do}^2 + \dot{y}_{do}^2 + \dot{z}_{do}^2)^2}, \\ &= \eta_2 \frac{v_d - v_o}{\|v_{do}\|^3}. \end{aligned} \tag{12}$$

Summing the velocities in (10) and (12) with the traditional controller (7) yields the full form of the extended potential field controller (ePFC), which is

$$v_d^{ePFC} = v_d^{PFC} - \lambda_2(v_d - v_t) + \sum_{i=0}^{n} \eta_2 \frac{v_d - v_o^i}{\|v_{do}^i\|^3}. \tag{13}$$

Finally, the velocity found in (13) must be transformed into the body coordinate system of the drone and is found to be

$$v_{d,body}^{ePFC} = v_d^{ePFC} * \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & sin(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

where ψ is the yaw angle of the drone around the body z axis.

To analyze the convergence of the proposed velocity controller (13) for the drone, we use the Lyapunov theory. We can choose a Lyapunov function as follows:

$$L = U_{att} = \frac{1}{2}\lambda_1 \|p_{dt}\|^2 + \frac{1}{2}\lambda_2 \|v_{dt}\|^2. \tag{15}$$

This function is positive definite, and the derivative of L is given by

$$\begin{aligned} \dot{L} &= \frac{1}{2}\lambda_1 \frac{\partial \|p_{dt}\|^2}{\partial p_{dt}} + \frac{1}{2}\lambda_2 \frac{\partial \|v_{dt}\|^2}{\partial v_{dt}}, \\ &= \lambda_1 \|v_{dt}\| + \lambda_2 \|a_{dt}\|, \end{aligned} \tag{16}$$

where a_{dt} is the relative acceleration between the drone acceleration and the target acceleration.

Note that the relative velocity between the drone and the target is designed following the direction of negative gradient of $U_{att}(p_{dt})$ with respect to p_{dt} as in (4). From (10), we can obtain

$$\begin{aligned} a_{dt} = \dot{v}_{dt} &= -\frac{1}{2}\lambda_2 \nabla^2(\dot{x}_{dt}^2 + \dot{y}_{dt}^2 + \dot{z}_{dt}^2), \\ &= \frac{1}{2}\lambda_2 \nabla_{v_{dt}}^2 \|v_{dt}\|^2, \\ &= -\lambda_2 \|\frac{a_{vt}(t) - a_{vt}(t-1)}{\Delta_t}\|, \end{aligned} \tag{17}$$

where Δ_t is a time step. Hence, substituting v_{dt} given by (4) and a_{dt} given by (17) into (16) we obtain

$$\dot{L} = -[\lambda_1^2 \|p_{dt}\| + \lambda_2^2 \|\frac{a_{vt}(t) - a_{vt}(t-1)}{\Delta_t}\|] \tag{18}$$

We can easily see that $\dot{L} < 0$ since $\|p_{dt}\|$ and $\|\frac{a_{vt}(t) - a_{vt}(t-1)}{\Delta_t}\|$ are positive. This mean that the proposed drone velocity controller is stable, or the drone is able to follow/track a moving target.

A comparison between the traditional controller and the extended form is provided in Section III using simulations. Section V presents the experimental results of implementing the ePFC on a drone in an actual testing environment.

III. Simulation

In order to validate the developed controller, the system was simulated using a Matlab Simulink model. The state space representation of the ARDrone's platform dynamics are take from the ARDrone Simulink Development Kit [37]. The complete Simulink model shown in Fig. 3 demonstrates how the ePFC controller uses feedback information from the ARDrone simulation and position estimator blocks. The output of the ARDrone simulation block is simply the velocity of the drone, and the position estimator uses an integrator with zero initial conditions to calculate position.

The desired path that the drone is to take is outlined in Table I. A virtual obstacle is placed at $(1, 1)$ which places it immediately in the path of the drone between waypoints 2 and 3. The drone is allowed two seconds at each waypoint in an attempt to let it settle before moving on to the next waypoint.

First, a traditional potential field controller was simulated, and the resulting path is shown in Fig. 4. The performance of the traditional PFC was poor as expected, because aerial drones have very different dynamics than their ground counterparts. Using the traditional PFC, the drone overshoots the

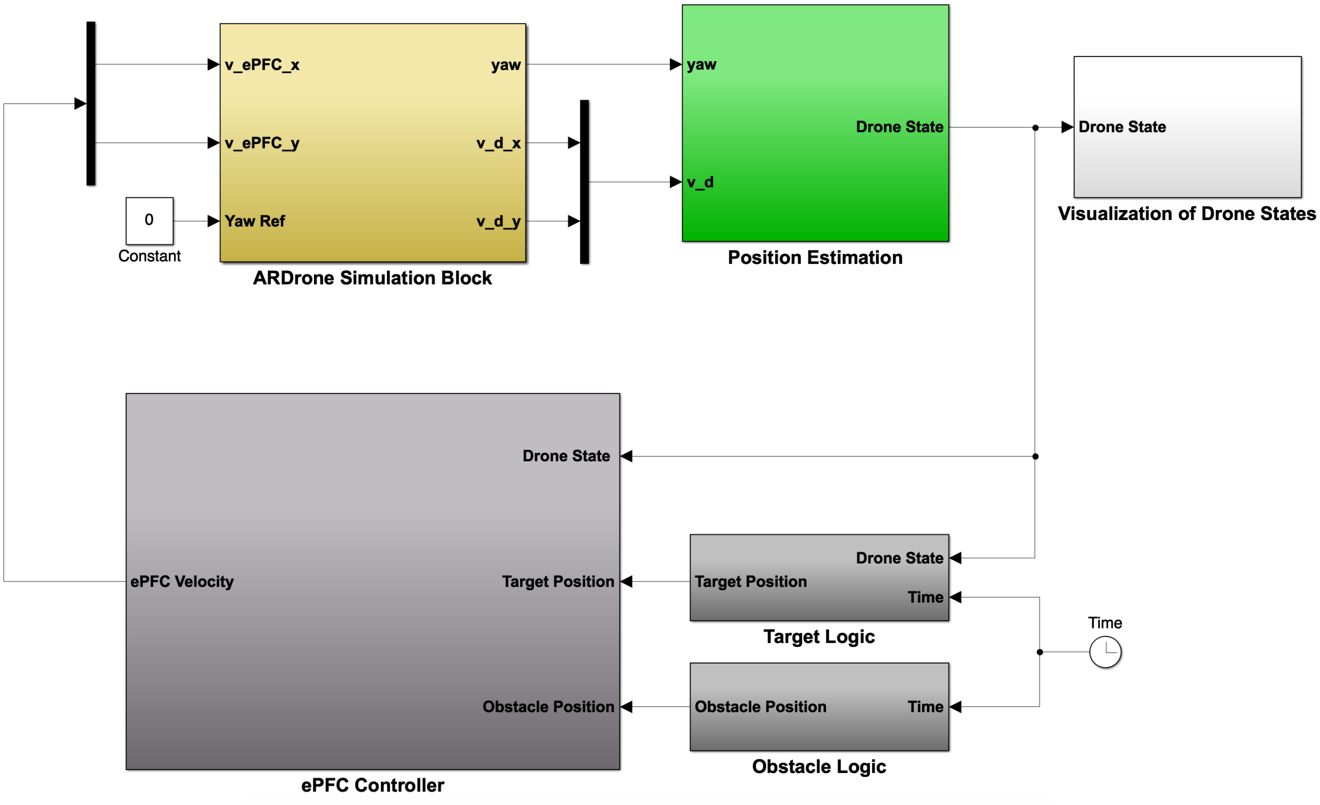

Fig. 3. The Simulink model used includes a state space representation from the ARDrone Simulink Development Kit, as well as custom blocks for the ePFC controller described in this work.

TABLE I
SIMULATION WAYPOINTS

Waypoint	X Coordinate [m]	Y Coordinate [m]
1	2.5	-1
2	2.5	1
3	-2.5	1
4	-2.5	-1

desired waypoint, and while it does avoid the obstacle at $(1, 1)$ it is not by much. The drone completed a full loop in approximately 35 seconds.

Next, the ePFC is tested using the same path and obstacle position. The results shown in Fig. 5 demonstrate the effectiveness of the new controller. The drone does not overshoot the desired waypoints and avoids the obstacle by a larger margin, while completing the course in a shorter amount of time than the traditional controller.

As outlined in Table II, the ePFC controller has zero overshoot, and has a settling time of approximately five seconds. This is a large improvement over the traditional controller which overshoots by up to 19% and takes nearly six seconds to settle. It is clear that the proposed controller is more appropriate for use on an aerial drone than the traditional PFC.

In addition to the comparison between the tradition PFC and the ePFC, a more complex simulation was performed which included several obstacles placed at random throughout the environment. The results shown in Fig. 6 demonstrate the that the ePFC is very effective in multi-obstacle scenarios, and it successfully navigates between waypoints without colliding

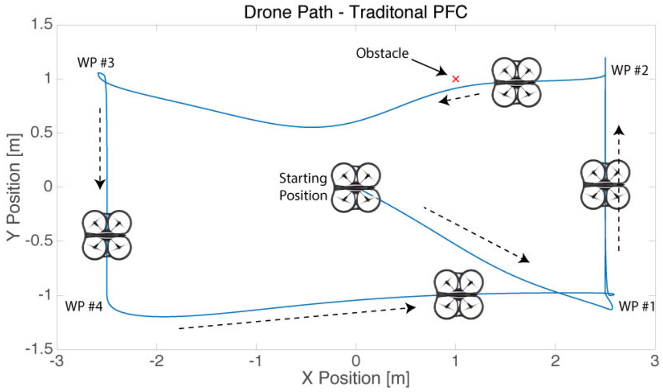

Fig. 4. A traditional PFC is simulated on the ARDrone, with poor results. Because drones cannot stop instantaneously like ground robots, the drone often overshoots the desired waypoint. For reference, the drone takes approximately 35 seconds to complete a full loop of the course.

TABLE II
SIMULATION CONTROLLER EVALUATION

Controller	Overshoot [%]	Settling Time [sec]
Traditional PFC	>19%	6
ePFC	0%	5

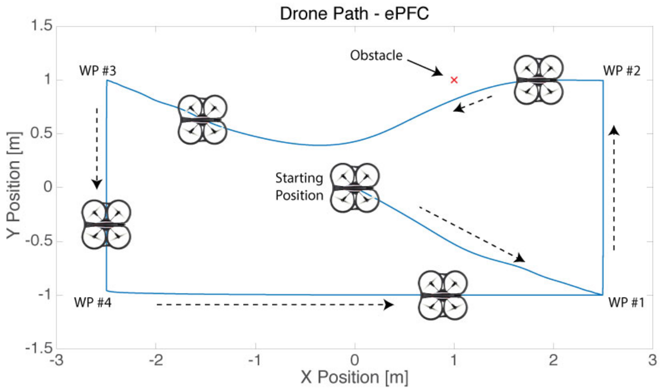

Fig. 5. Using the extended potential field controller (ePFC), the drone is able to complete the course without overshooting the target waypoints, and avoids the obstacle by a larger margin than the traditional controller. Because the drone does not overshoot the target, it is able to complete the course in a shorter amount of time compared to the traditional controller.

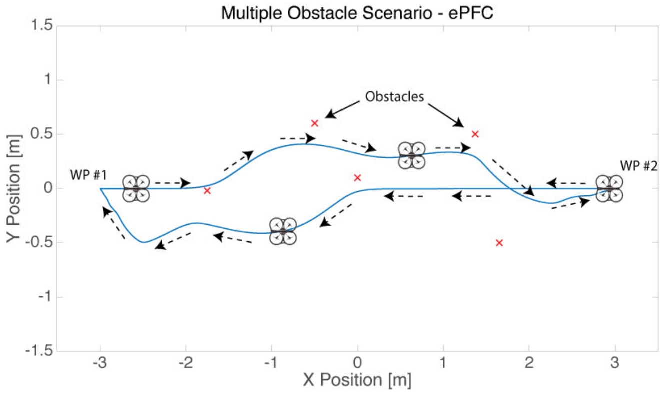

Fig. 6. A more complex simulation was performed with multiple obstacles placed throughout the environment. The drone is able to successfully navigate between waypoints without colliding with a single obstacle, thus demonstrating its effectiveness in multi-obstacle scenarios.

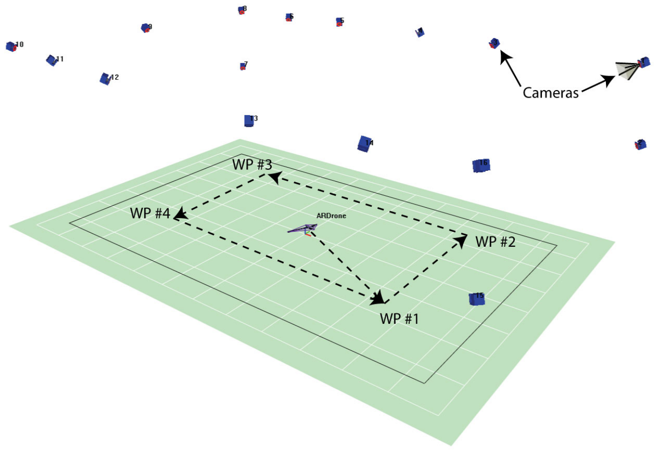

Fig. 7. The Motion Analysis Cortex software gives the user a real-time, visual 3D representation of the environment including camera locations and any objects sensed by the system [38].

with a single obstacle.

IV. Experimental Setup

The experimental platform chosen to implement the ePFC is the ARDrone 2.0 quadrotor. This platform was chosen for its ease of communication - over a wifi connection - as well as the safety provided by the foam hull. Additionally, the ARDrone requires little to no setup and spare parts are readily available in case of crashes. The ARDrone 2.0 can be equipped with a 1500 mAh battery which yields flight times up to 18 minutes. Large batteries and long flight times are very advantageous in a testing environment because it allows for more uninterrupted tests and less downtime recharging batteries. The ARDrone 2.0 is also equipped with a 1GHz 32 bit ARM Cortex A8 processor, 1 GB DDR2 RAM, and runs Linux. This means that the developed controller can be implemented onboard the drone in future work.

Sixteen Motion Analysis Kestrel cameras located throughout the testing space provide the position and orientation of the drone, target (if not virtual), and any obstacles present. The Cortex software suite provides a visual representation of the environment as shown in Fig. 7 as well as sending data over a network connection for use by external programs.

The overview of the experimental setup shown in Fig. 8 demonstrates the feedback loop implemented. The Motion Analysis external tracking system is used for external localization of the drone and obstacles in real time. The position information is used by the same Simulink model shown in Section III which controls the ARDrone over a wireless connection.

V. Experimental Results

Having validated the controller using the Simulink simulation, it was then implemented on the actual ARDrone. Because the simulation showed a clear improvement in performance between the tradition PFC and the developed ePFC, the traditional controller was not tested on the experimental platform. Instead, the ePFC was immediately used to perform the a similar experiment to the one which was simulated. The general path defined by waypoints in Table I is implemented, however due to size limitations of the testing space, the rectangular path is smaller than the simulated one.

The results from the experiment shown in Fig. 9 and Fig. 10 demonstrate that the drone successfully reaches each waypoint, and also avoids the obstacle in its path between waypoints two and three.

To quantify the controller performance, the error in response to a waypoint change, or step input, is shown in Fig. 11. The X axis error is chosen as the worst case scenario in the experiment, having a step input of over 2.5 meters versus only 1 meter on the Y axis.

The controller's performance is quite good to step inputs, with an approximate settling time of 5.5 seconds, and a percent overshoot of only 1.8%. A comparison between the simulated and experimental results is outlined in Table III. While the experimental results do have a slightly longer settling time, and more overshoot, this is not surprising. In a real world application, the controller is subject to disturbances such as ground effects from propeller wash, since the drone is operating close to the ground and desks.

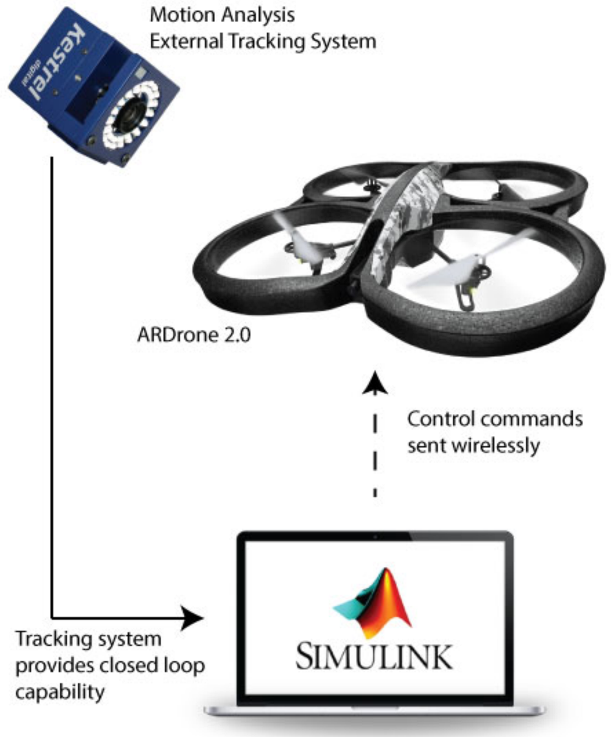

Fig. 8. The experimental setup for this work includes a Motion Analysis external tracking system with 16 cameras which provides the position of the drone and obstacles in the environment. The same Simulink model used in Section III provides control commands to the ARDrone over a wireless connection.

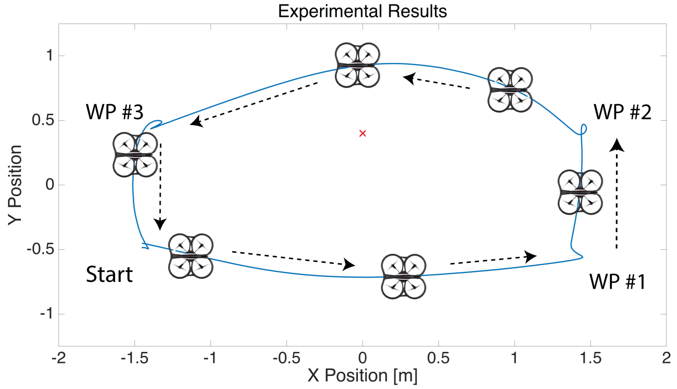

Fig. 9. To test the ePFC experimentally, the drone follows waypoints similar to those in the simulation. The drone successfully reaches each waypoint and avoids the obstacle in its path between waypoints two and three.

Having evaluated the performance of the drone's ability to track targets, the obstacle avoidance functionality must be addressed as well. As the drone approaches the obstacle, the repulsive potential pushes the drone around it as expected. In this experiment, the drone avoids the obstacle by a margin of approximately 8 inches. If a larger margin is required to accommodate and larger obstacle or drone, the repulsive potential scale factor, η_1 can simply be increased which has the effect of making the repulsive field wider.

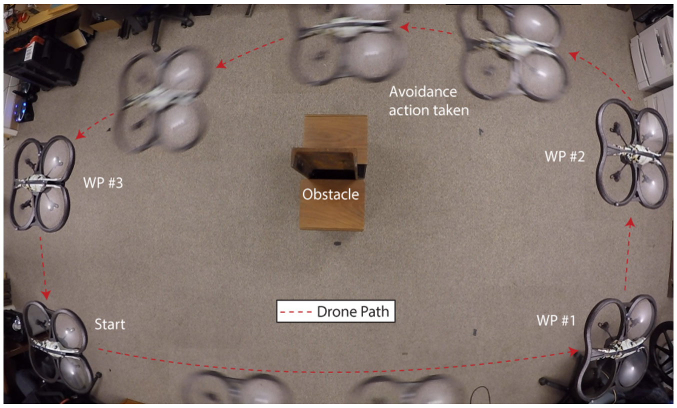

Fig. 10. Stills from a video demonstrate the drone taking avoidance action while tracking moving waypoints.

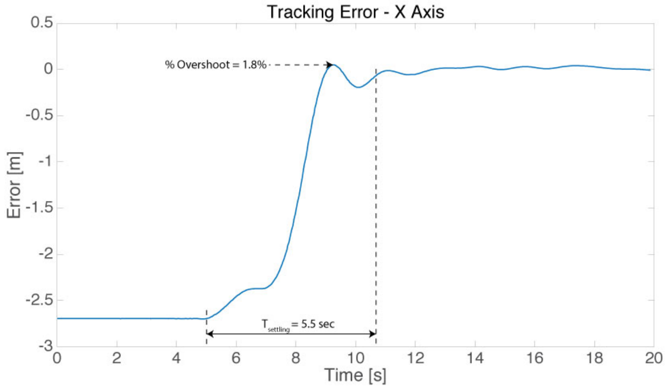

Fig. 11. The error in response to a waypoint change, or a step input, results in a settling time of approximately 5.5 seconds and a percent overshoot of only 1.8%. The X axis was chosen because the step input for this direction was the largest, at over 2.5 meters, whereas the Y input is only 1 meter.

VI. Conclusions

This paper presented an extended potential field controller (ePFC) which augments the traditional PFC with the capability to use relative velocities between a drone and a target or obstacles as feedback for control. Next, the stability of the ePFC was proven using Lyapunov methods. Additionally, the presented controller was simulated and its performance relative to a tradition PFC was evaluated. The evaluation shows that the ePFC performs significantly better than a traditional PFC by reducing overshoot and settling time when navigating between waypoints. Finally, experimental results were presented which showed the actual performance of the controller.

Future work may include using an experimental system with completely onboard sensing capabilities. Potentially, the front facing camera on the ARDrone 2.0 could be used for

TABLE III
Simulation vs Experimental Evaluation

Experiment	Overshoot [%]	Settling Time [sec]
Simulated ePFC	0%	5
Experimental ePFC	1.8%	5.5

localization using computer vision algorithms. Additionally, the controller may be implemented onboard the drone itself. Because the ePFC presented is not computationally intensive, it can be implemented on nearly any drone on the market today.

References

[1] J. M. de Dios, L. Merino, F. Caballero, A. Ollero, and D. Viegas, "Experimental results of automatic fire detection and monitoring with uavs," *Forest Ecology and Management*, vol. 234, Supplement, pp. S232 –, 2006.

[2] L. Merino, F. Caballero, J. Martnez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[3] D. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. Mehra, "Forest fire monitoring with multiple small uavs," in *American Control Conference, 2005. Proceedings of the 2005*, June 2005, pp. 3530–3535 vol. 5.

[4] P. Sujit, D. Kingston, and R. Beard, "Cooperative forest fire monitoring using multiple uavs," in *Decision and Control, 2007 46th IEEE Conference on*, Dec 2007, pp. 4875–4880.

[5] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 639–643.

[6] R. Pitre, X. Li, and R. Delbalzo, "Uav route planning for joint search and track missions: An information-value approach," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 48, no. 3, pp. 2551–2565, July 2012.

[7] A. Birk, B. Wiggerich, H. Bulow, M. Pfingsthorn, and S. Schwertfeger, "Safety, security, and rescue missions with an unmanned aerial vehicle (uav)," *Journal of Intelligent and Robotic Systems*, vol. 64, no. 1, pp. 57–76, 2011.

[8] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *Robotics Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, Sept 2012.

[9] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini uav," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.

[10] D. Erdos, A. Erdos, and S. Watkins, "An experimental uav system for search and rescue challenge," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 28, no. 5, pp. 32–37, May 2013.

[11] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked uavs as aerial sensor network for disaster management applications," *Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[12] I. Maza, F. Caballero, J. Capitan, J. Martinez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 563–585, 2011.

[13] P. Bupe, R. Haddad, and F. Rios-Gutierrez, "Relief and emergency communication network based on an autonomous decentralized uav clustering network," in *SoutheastCon 2015*, April 2015, pp. 1–8.

[14] H. La, R. Lim, B. Basily, N. Gucunski, J. Yi, A. Maher, F. Romero, and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1655–1664, Dec 2013.

[15] S. J. Mills, J. J. Ford, and L. Mejias, "Vision based control for fixed wing uavs inspecting locally linear infrastructure using skid-to-turn maneuvers," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1-4, pp. 29–42, 2011.

[16] Z. Li, Y. Liu, R. Walker, R. Hayward, and J. Zhang, "Towards automatic power line detection for a uav surveillance system using pulse coupled neural filter and an improved hough transform," *Machine Vision and Applications*, vol. 21, no. 5, pp. 677–686, 2010.

[17] R. Bloss, "Unmanned vehicles while becoming smaller and smarter are addressing new applications in medical, agriculture, in addition to military and security," *Industrial Robot: An International Journal*, vol. 41, no. 1, pp. 82–86, 2014.

[18] J. Roldan, G. Joossen, D. Sanz, J. del Cerro, and A. Barrientos, "Mini-uav based sensory system for measuring environmental variables in greenhouses," *Sensors*, vol. 15, no. 2, pp. 3334–3350, 2015.

[19] M. Rossi, D. Brunelli, A. Adami, L. Lorenzelli, F. Menna, and F. Remondino, "Gas-drone: Portable gas sensing system on uavs for gas leakage localization," in *SENSORS, 2014 IEEE*, Nov 2014, pp. 1431–1434.

[20] M. Shaohua, X. Jinwu, and L. Zhangping, "Navigation of micro aerial vehicle in unknown environments," in *25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 322–327.

[21] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[22] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 20–25.

[23] I. Sa and P. Corke, "System identification, estimation and control for a cost effective open-source quadcopter," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 2202–2209.

[24] A. Kendall, N. Salvapantula, and K. Stol, "On-board object tracking control of a quadcopter with monocular vision," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 404–411.

[25] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3d exploration with a micro-aerial vehicle," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 9–15.

[26] J.-E. Gomez-Balderas, P. Castillo, J. Guerrero, and R. Lozano, "Vision based tracking for a quadrotor using vanishing points," *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1-4, pp. 361–371, 2012.

[27] L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[28] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2520–2525.

[29] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, Sept 2010.

[30] A. Dogan, "Probabilistic approach in path planning for uavs," in *Intelligent Control. 2003 IEEE International Symposium on*, Oct 2003, pp. 608–613.

[31] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning," in *Proceedings of the AIAA guidance, navigation and control conference*, 2005.

[32] A. A. Neto, D. Macharet, and M. Campos, "Feasible path planning for fixed-wing uavs using seventh order bezier curves," *Journal of the Brazilian Computer Society*, vol. 19, no. 2, pp. 193–203, 2013.

[33] A. Altmann, M. Niendorf, M. Bednar, and R. Reichel, "Improved 3d interpolation-based path planning for a fixed-wing unmanned aircraft," *Journal of Intelligent and Robotic Systems*, vol. 76, no. 1, pp. 185–197, 2014.

[34] H. M. La, R. S. Lim, W. Sheng, and J. Chen, "Cooperative flocking and learning in multi-robot systems for predator avoidance," in *IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)*, May 2013, pp. 337–342.

[35] H. M. La and W. Sheng, "Multi-agent motion control in cluttered and noisy environments," *Journal of Communications*, vol. 8, no. 1, p. 32, 2013.

[36] ——, "Dynamic target tracking and observing in a mobile sensor network," *Robotics and Autonomous Systems*, vol. 60, no. 7, p. 996, 2012.

[37] D. E. Sanabria. Ardrone simulink development kit v1.1. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/43719-ar-drone-simulink-development-kit-v1-1

[38] Motion analysis systems. [Online]. Available: http://www.motionanalysis.com