IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2017, accepted January 26, 2018, date of publication February 5, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2802468

Mobile Robot Rendezvous Using Potential Fields combined With Parallel Navigation

PATRICK FRIUDENBERG AND SCOTT KOZIOL, (Member, IEEE)
Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798, USA
Corresponding author: Scott Koziol (scott_koziol@baylor.edu)

ABSTRACT Industrial robotics, military, surveying, and delivery applications have laid a foundation for research into autonomous machines, including unmanned aerial vehicles, autonomous ground vehicles, and autonomous undersea vehicles. This paper supports this research by developing a new guidance method which combines potential fields, typically used for obstacle avoidance, and parallel navigation, a popular missile guidance method. The new unified algorithm allows a mobile robot interceptor to guide to, and rendezvous with, a moving target while avoiding obstacles in its path. A Matlab simulator is used to analyze performance for different interceptor-target geometries and obstacle densities. Simulations and analysis show that the combined algorithm increases performance by reducing the time-to-contact in most of the cases. Specifically, it was found to provide a 14.4%-21.3% improvement in approximately 96%-98% of the simulation cases.

INDEX TERMS Guidance, mobile robots, parallel navigation, path planning, potential fields, robots.

I. INTRODUCTION

It is often necessary for a mobile robot to intercept and rendezvous with a moving target. ''Rendezvous is the guidance when, in addition to the coincidence of an object's and a target's position, the object's velocity equals the target velocity [1].'' Sometimes this scenario takes place in the presence of moving or static obstacles. A simple example is a robot shepherding task where a ''robot sheepdog'' must rendezvous with a ''robot sheep'' which is going off course. The ''sheepdog'', at the same time, must avoid moving obstacles like other robot shepherds, sheep, or static obstacles like fence poles [2]. Existing approaches for guiding a robot or a missile to a moving target have used fuzzy logic [3], Line of Sight (LOS) guidance laws [4], [5], Visual Servoing [6], Proportional Navigation [7]–[10], Pursuit and Proportional Navigation [11], and Potential Fields [12]–[16]. Planning laws have also be designed to optimize detection probability [16]. Potential Fields has been used for rendezvous which incorporates obstacle avoidance [15]. In that work the duration of the maneuver was not considered as a critical-parameter. As a result, the Terminal Phase (i.e. the final meters of the asymptotic maneuver) required most of the time.

Individual guidance methods have been combined for a hybrid approach to overcome difficult scenarios or shortcomings of the individual methods. Proportional Navigation, for example, has been combined with integral components to compensate for the difficult scenario of maneuvering targets [17]. Potential fields, while effective, has a well-known shortcoming of local minima problems [4], [18]. To overcome this, it has been coupled with techniques such as fuzzy logic [19] and genetic algorithms [20]. Potential Fields has also been combined with sliding mode control [21], with fuzzy logic [19], with stochastic reachable sets [22], and with Navigation Templates [23]. Obstacle avoidance is not typically included in Proportional Navigation theory and application, however previous research demonstrated that it is possible [4], [24]. Other researchers have developed solutions for rendezvous in the presence of obstacles. Kunwar *et al* [26], developed a LOS method augmented with a velocity change maneuver to avoid obstacles [27], and they further improved upon it to improve time optimality [27], [28].

The main contribution of this work is a method combining Parallel Navigation (PN) and Potential Fields (PF) guidance methods to add obstacle avoidance to Parallel Navigation, and improve overall performance metrics such as a reduction in rendezvous time [29].

In the combined guidance law presented in this paper, the interceptor robot's velocity is controlled by a vector using aspects of both Parallel Navigation and Potential Fields. The vector's *direction* is determined using Parallel Navigation and

the vector's *magnitude* is determined using Potential Fields. This gives the robot interceptor the ability to rendezvous with a moving target while avoiding obstacles. As with most guidance techniques, our method relies on the interceptor robot's sensors or an external observer to obtain an accurate estimate of the target's state [30]. This estimation is often accomplished with Kalman filters. The focus of this paper is guidance and not state estimation so we assume that appropriate state estimates of the robot, target, and obstacles already exist.

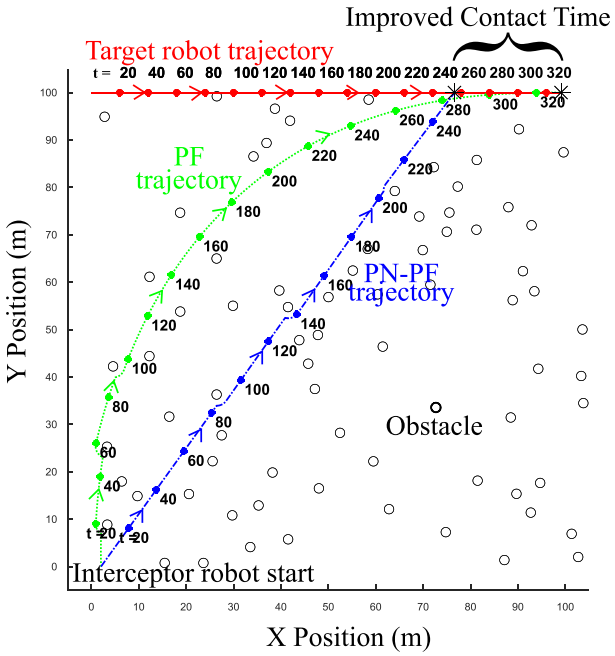

FIGURE 1. Big Picture: The Parallel Navigation-Potential Fields (PN-PF) guidance algorithm developed in this paper is compared against the Potential Fields (PF) algorithm. The goal is to reduce the time needed to rendezvous with a moving robot in the presence of obstacles in a two dimensional environment.

A picture illustrating our paper is found in Fig. 1. Our algorithm is called the Parallel Navigation-Potential Fields (PN-PF) guidance algorithm. Fig. 1 shows a target moving across the top to the right. The interceptor starts at the bottom left. The trajectories of two algorithms are shown, and the goal is to make contact quickly with the target. The PN-PF trajectory (blue line) makes contact with the target before the PF trajectory (green line). This rendezvous takes place in an environment with randomly placed static obstacles.

The rest of this paper is organized as follows. Section II describes background about Potential Fields and Parallel Navigation. Section III describes our combined algorithm. Section IV describes three sets of experimental results and analysis comparing PN-PF to PF, and Section V summarizes and presents the final conclusions.

II. BACKGROUND

Background on Potential Fields (PF) and Parallel Navigation (PN) is given in the next two subsections.

A. POTENTIAL FIELDS

Vectors and vector fields have been used to control UAVs [31]–[33], nonholonomic wheeled robots [34], [35], and robotic arms [36]. The use of vector fields to direct a UAV on both straight and circular paths was developed and proven to be stable using the Lyaponuv method [37]–[39]. The Potential Fields approach uses two components to produce a control vector [36]. The attractive component, U_{att}, guides the interceptor robot to the intended target, while the repellent component, U_{rep}, guides the interceptor robot away from obstacles. Together they form the total artificial potential energy function, (1) [36].

$$U_{art} = U_{att} + U_{rep} \tag{1}$$

The attractive component (2) is due to the target [36].

$$U_{att} = \frac{1}{2} k_{att} \left(\left| \vec{t} - \vec{r} \right| \right)^2 \tag{2}$$

In (2), $\vec{t}$ is the position of the target and $\vec{r}$ is the position of the robot interceptor. These vectors are with respect to an inertial navigation reference frame. Finally, k_{att} is a designer controlled gain. U_{att} is minimized when the target and interceptor are co-located, therefore we want to minimize U_{att} for our rendezvous scenario. To allow the interceptor's guidance law to minimize (2), its gradient is used to create a vector field telling the interceptor what direction to move, (3) [36]. This velocity control vector is $\dot{\vec{r}}_{att}$, (3) [36].

$$\dot{\vec{r}}_{att} = \nabla U_{att} = k_{att} \left(\vec{t} - \vec{r} \right) \tag{3}$$

If the target is in motion then (3) shows that the interceptor robot would not rendezvous with the target. This is because (3) goes to 0 when the interceptor robot reaches the target, therefore, $\dot{\vec{r}}_{att} \rightarrow 0$ as the interceptor approaches the target, so at the point of coincidence, the interceptor's velocity is 0, but the target's velocity is not. This is not a rendezvous because their velocities are not matched [1]. This demonstrates what Lyapunov's theorem also shows: that this method, without some modification, only ensures convergence in position and not convergence in velocity [15]. One way this velocity convergence problem has been solved is by carefully choosing the control function, however the final stages of velocity matching can take a lot of time [15]. One of this paper's main contributions is a solution to this problem which reduces the time-to-intercept. When rendezvousing with a moving target, the magnitude in (3) must be modified to account for the velocity of the target. The attractive potential function, U_{att}, is adjusted to include the speed of the target, $\dot{\vec{t}}$, (4).

$$U_{att} = \frac{1}{2} k_{att} \left(\left| \vec{t} - \vec{r} \right| \right)^2 + \underbrace{\frac{1}{2} k_{vel} \left| \dot{\vec{t}} \right|^2}_{\text{Rendezvous Term}} \tag{4}$$

The new energy equation for the attractive portion of the potential changes the vector field associated with it (5).

$$\nabla U_{att} = k_{att} \left(\vec{t} - \vec{r} \right) + k_{vel} \dot{\vec{t}} \tag{5}$$

If the target is in motion then (5) shows that the velocities will match and the interceptor robot will rendezvous with the target [1]. The repelling component of the artificial field, U_{rep}, is expressed as (6) where $\vec{o}_i$ represents the position of obstacle i, ρ denotes the radius of repulsion defined by the designer, and k_{rep} is a user defined control gain [36]. The summation accounts for repulsive force of each obstacle because the total repulsive factor of the field is the sum of all the individual obstacle's repulsive fields.

$$U_{rep} = \begin{cases} \sum_i \frac{1}{2} k_{rep} \left(\frac{1}{|\vec{r} - \vec{o}_i|} - \frac{1}{\rho} \right)^2, & |\vec{r} - \vec{o}_i| \leq \rho \\ 0, & otherwise \end{cases} \tag{6}$$

The gradient of (6) provides a vector valued function to provide the interceptor robot with directional travel information.

$$\nabla U_{rep} = \begin{cases} \sum_i k_{rep} \left(\frac{1}{|\vec{r} - \vec{o}_i|} - \frac{1}{\rho} \right) \left(\frac{\vec{r} - \vec{o}_i}{|\vec{r} - \vec{o}_i|^3} \right), & |\vec{r} - \vec{o}_i| \leq \rho \\ 0, & otherwise \end{cases} \tag{7}$$

In (7), $\vec{r} - \vec{o}_i$ in the numerator is a vector pointing away from an obstacle in the direction of the interceptor robot, and the $|\vec{r} - \vec{o}_i|$ term in the denominator makes the magnitude grow inversely proportional to the distance between the interceptor robot and the obstacle. This is the behavior expected from the velocity of the interceptor, so it will be used as a velocity control vector, $\dot{\vec{r}}_{rep}$ (8) .

$$\dot{\vec{r}}_{rep} = \nabla U_{rep} \tag{8}$$

The total artificial potential field affecting the interceptor robot, U_{art}, is determined by substituting (4) and (6) into (1). To guide the interceptor robot to the target while simultaneously avoiding obstacles one takes the gradient of U_{art} to produce a vector field that points away from the objects and towards the target and can be used as a velocity control vector.

$$\nabla U_{art} = \nabla U_{att} + \nabla U_{rep} \tag{9}$$

This leads to a final velocity control vector, $\dot{\vec{r}}_{ref}$, from (5) and (8).

$$\dot{\vec{r}}_{ref} = \dot{\vec{r}}_{att} + \dot{\vec{r}}_{rep} \tag{10}$$

B. PARALLEL NAVIGATION

Parallel Navigation is an ancient guidance rule used by mariners [1]. To determine the direction the interceptor robot should take to insure contact with a moving target, the interceptor can construct a velocity control vector by matching the target's velocity in the direction normal to the Line of Sight (LOS) and then applying whatever power is left in the interceptor along the closing velocity direction, Fig. 2.

If the interceptor and target robots have equal velocity components normal to the LOS, and the component of the interceptor's velocity parallel to the line of sight is pointed at the target, then contact will be made at some future time.

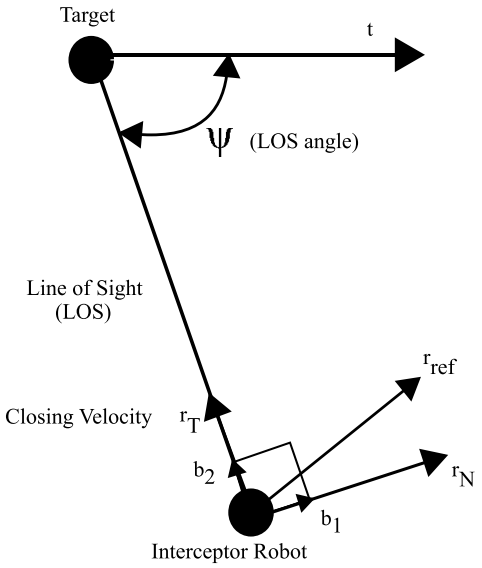

FIGURE 2. Parallel Navigation: This is a planar representation of intercept geometry [1]. If r_T is aligned with the LOS then ψ will remain constant throughout the time-to-intercept. If r_T is not aligned with the LOS then the LOS will rotate and the vehicles will not intercept unless the interceptor makes a maneuver [40].

This reduces the complexity of the problem to performing a change of basis on the target's velocity vector to get parallel and normal components to the LOS. Fig. 2 shows a planar representation of the Parallel Navigation intercept geometry. It shows two unit basis vectors: b_1 pointing normal to the LOS, and b_2 pointing parallel to the LOS.

Our goal in this section, on our way to the proposed combined algorithm, is to express the interceptor's velocity vector, $\dot{\vec{r}}_{ref}$, in terms of a maximum interceptor velocity (a physical constraint) and the geometry of the engagement. First, it is described in terms of components normal and parallel to the LOS:

$$\dot{\vec{r}}_{ref} = \dot{\vec{r}}_T + \dot{\vec{r}}_N \tag{11}$$

We then express the target's velocity relative to the two unit basis vectors.

$$\dot{\vec{t}} = \left(\text{Proj}_{\vec{b}_1} \dot{\vec{t}} \right) + \left(\text{Proj}_{\vec{b}_2} \dot{\vec{t}} \right) \tag{12}$$

This allows us to solve for the target's velocity in the b_1 direction:

$$\text{Proj}_{\vec{b}_1} \dot{\vec{t}} = \dot{\vec{t}} - \text{Proj}_{\vec{b}_2} \dot{\vec{t}} \tag{13}$$

From this, we construct a velocity control vector normal to the LOS for the interceptor:

$$\dot{\vec{r}}_N = \dot{\vec{t}} - \left(\text{Proj}_{\vec{b}_2} \dot{\vec{t}} \right) \tag{14}$$

Where:

$$\text{Proj}_{\vec{b}_2} \dot{\vec{t}} = \left(\dot{\vec{t}} \cdot \vec{b}_2 \right) \vec{b}_2 = \left(\vec{b}_2 \vec{b}_2^T \right) \dot{\vec{t}} \tag{15}$$

And:

$$\vec{b}_2 = \frac{LOS}{|LOS|} = \frac{\vec{t} - \vec{r}}{|\vec{t} - \vec{r}|} \tag{16}$$

Which leads to one of the components of the interceptor's velocity control vector:

$$\dot{\vec{r}}_N = \dot{\vec{t}} - \left(\vec{b}_2\vec{b}_2^T\right)\dot{\vec{t}} = \left(I - \vec{b}_2\vec{b}_2^T\right)\dot{\vec{t}} \tag{17}$$

One can re-arrange (11) to solve for the magnitude of the closing velocity, $\dot{\vec{r}}_T$, in terms of the reference and normal velocity components (18).

$$\left|\dot{\vec{r}}_T\right| = \sqrt{\left|\dot{\vec{r}}_{ref}\right|^2 - \left|\dot{\vec{r}}_N\right|^2} \tag{18}$$

Assuming the interceptor applies its maximum velocity for $\dot{\vec{r}}_{ref}$ (19),

$$\left|\dot{\vec{r}}_{ref}\right| \equiv \dot{r}_{\max} \tag{19}$$

then one can substitute (19) into (18) to solve for the magnitude of the closing velocity, $\dot{\vec{r}}_T$, in terms of the maximum and normal velocity components (20).

$$\left|\dot{\vec{r}}_T\right| = \sqrt{\dot{r}_{\max}^2 - \left|\dot{\vec{r}}_N\right|^2} = \sqrt{\dot{r}_{\max}^2 - \dot{\vec{r}}_N^T\dot{\vec{r}}_N} \tag{20}$$

The final velocity control vector can be expressed as a function of the interceptor's max velocity and the geometry of the engagement (21).

$$\dot{\vec{r}}_{ref} = \left(\sqrt{\dot{r}_{\max}^2 - \dot{\vec{r}}_N^T\dot{\vec{r}}_N}\right)\vec{b}_2 + \dot{\vec{r}}_N \tag{21}$$

III. COMBINING PARALLEL NAVIGATION AND POTENTIAL FIELDS

The control law for the proposed Parallel Navigation-Potential Fields (PN-PF) hybrid algorithm is composed of attractive and repelling components. The attractive component is based on Parallel Navigation (PN) equation (21) where the Potential Fields (PF) attractive component in (5) is used to set the magnitude of the PN reference vector with the restriction that the PF attractive component is less than or equal to the interceptor robot's maximum velocity (22).

$$|\nabla U_{att}| = \left|k_{att}\left(\vec{t} - \vec{r}\right) + k_{vel}\dot{\vec{t}}\right| \leq \left|\dot{\vec{r}}_{max}\right| \tag{22}$$

We substitute $|\nabla U_{att}|$ for $\dot{r}_{\max}$ in (21) and assign the resulting equation (23) to be the attractive portion of the final PN-PF component.

$$\dot{\vec{r}}_{att} = \left(\sqrt{|\nabla U_{att}|^2 - \vec{r}_N^T\vec{r}_N}\right)\vec{b}_2 + \dot{\vec{r}}_N \tag{23}$$

Where $\dot{\vec{r}}_N$ in (23) is from (17). This attractive component therefore changes according to the angle between the target and interceptor, and the rendezvous (intercept) distance between them. This is because r_N, b_2, and ∇U_{att} are functions of the rendezvous geometry. The repelling component of the PN-PF guidance is assigned to the standard PF repelling function, (8). The final PN-PF control law is obtained by substituting (23) and (8) into (24).

$$\dot{\vec{r}}_{refPNPF} = \dot{\vec{r}}_{att} + \dot{\vec{r}}_{rep} \tag{24}$$

Pseudocode for implementing the PN-PF guidance algorithm is shown in Algs 1, 2, 3, and 4. The highest level pseudocode is Alg 1. It contains constants describing the obstacles and the interceptor. *ORAD* is the radius of the obstacles, and *RRAD* is the radius of the interceptor robot. It first calls Alg 2, the PF Attractor Algorithm, to find the attractive component ∇U_{att}, (22), which will be used to set the magnitude of the PN reference vector. It has two user-defined control gain constants K_{ATT} and K_{VEL}. The high level pseudocode then passes this newly calculated number to Alg 3, the PN Navigation Algorithm, to calculate (23). The high level pseudocode then repeatedly calls Alg 4, the Potential Field Repelling Algorithm, to calculate the PF repelling vectors for each obstacle (7-8). Finally, the high level pseudocode computes the proposed PN-PF control law with a simple addition of terms, (24).

Algorithm 1 Main PN-PF Algorithm

```
procedure PF–PN
    ∇U_att = POT_FIELD_ATT(r, t, tdot, K_ATT, K_VEL)
    V_att = PAR_NAV(r,t,tdot,|∇U_att|)
    V_rep = 0
    for each i i ∈ Num_of_obstacles do
        V_r = POT_FIELD_REP(r, RRAD, o_i, ORAD, k_rep, RHO)
        V_rep = V_rep + V_r
    end for
    V_refPNPF = V_att + V_rep
    if |V_refPNPF| > ṙ_max then
        V_refPNPF = ṙ_max V_refPNPF / |V_refPNPF|
    end if
end procedure
```

Algorithm 2 Potential Field Attractor Algorithm

```
procedure POT_FIELD_ATT
    ∇U_att = K_ATT (t − r) + K_VEL (tdot)
end procedure
```

Algorithm 3 Parallel Navigation Algorithm

```
procedure PAR_NAV(r,t,tdot,rdot_mag)
    b = (t − r) / |t − r|
    rdot_N = (eye(length(r)) − b ∗ b′) ∗ tdot
    rdot_T = sqrt(rdot_mag^2 - rdot_N′ ∗ rdot_N) ∗ b
    v_ref = rdot_T + rdot_N return v_ref
end procedure
```

IV. RESULTS AND ANALYSIS

This section describes the performance of the proposed PN-PF algorithm based upon Matlab simulations. Mobile robots have both dynamic and kinematic constraints. Our simulations assume a holonomic kinematic model with the

Algorithm 4 Potential Field Repelling Algorithm

procedure POT_FIELD_REP
  $o = o + ORAD * ((r - o)/norm(r - o))$
  $r = r + RRAD * ((o - r)/norm(o - r))$
  $los = r - o$
  $mlos = norm(r - o)$
  if $mlos < RHO$ **then**
    $\nabla U_{rep} = K_{REP}\left(\frac{1}{mlos} - \frac{1}{RHO}\right)\left(\frac{los}{mlos^3}\right)$
  else
    $\nabla U_{rep} = zeros(length(r), 1)$
  end if return ∇U_{rep}
end procedure

ability to perform high speed movements to respond to the control commands. Future studies planned involve adding a dynamic model and nonholonomic constraints like in a differential drive robot [41].

The simulations test the PN-PF algorithm by evaluating its performance sensitivity to 1) obstacle density and 2) interceptor-target engagement geometry. To quantify performance of our proposed guidance algorithm we compare PN-PF guidance to PF guidance alone. The performance metric we utilize is the time-to-contact with the target, while insuring no obstacles are contacted. Three sets of results are presented: 1) Uniformly distributed obstacles with defined interceptor starting positions (Sec IV-A), 2) Randomly distributed obstacles with defined interceptor starting positions (Sec IV-B), and 3) Randomly distributed obstacles with random interceptor starting positions (Sec IV-C). The results characterize the algorithm's performance over varying obstacle density and rendezvous engagement scenarios.

A. RESULTS 1: UNIFORMLY PLACED OBSTACLES AND DEFINED INTERCEPTOR STARTING LOCATIONS

In these scenarios the target moves along the top of the environment from left to right, and the interceptor robot is placed at varied locations along the bottom of the environment, Fig. 3.

The obstacles were placed in a 100 X 100 m area. The obstacles each have a radius of 1 meter. The pseudocode setting for this dimension is $ORAD = 1$. For each set of obstacles, the course was executed 50 times while changing the initial position of the interceptor robot. Its initial position was placed at equally spaced locations along the x axis (≈2m spacing). The test scenario is shown in Fig. 3. Once the 50 scenarios were ran for both PN-PF and PF the number of objects in the field was changed and another 50 scenarios were run. A total of 1200 scenarios were executed: 600 with PF and 600 with PN-PF. Fig. 4 show a graphic of the test organization. The following gain constants were used: $K_{vel} = 1$, $K_{rep} = 15$, $K_{att} = 4$, $RHO = 1.25$. The time step size used in the simulations is 0.05 s. If the Interceptor was within 0.05 m of the target in the simulations it was considered a rendezvous. The interceptor robot has a radius

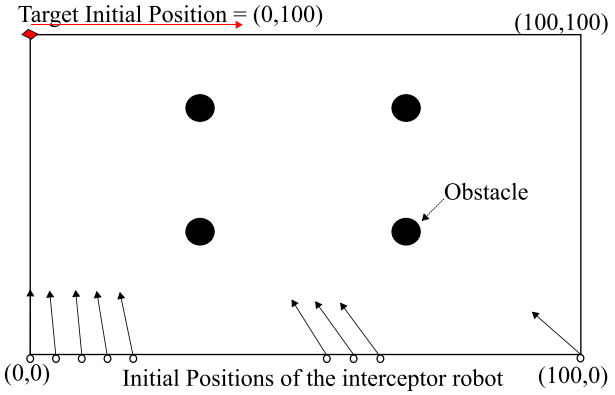

FIGURE 3. Results 1: Matlab test scenario Setup. For each test, the target robot always started at position(0,100) and moved towards position (100,100) with a constant velocity. This picture shows an example obstacle scenario of 4 obstacles uniformly placed in the environment. Fifty tests for this obstacle density would be run for Results Set 1. For each of the fifty tests, the interceptor robot's starting position was one of the fifty starting locations along the axis (0,0) to (100,0).

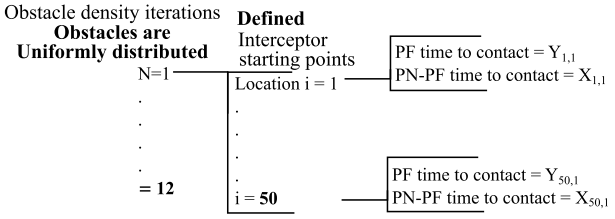

FIGURE 4. Results 1: Data tree for the Matlab experiments which evaluated performance as a function of obstacle density. In these cases, the obstacles were uniformly placed in the environment, and the interceptor robot started at 50 pre-defined locations.

of 0.1651 meters ($RRAD = 0.1651$) and has a maximum velocity of 0.5m/s ($\dot{r}_{max} = 0.5$).

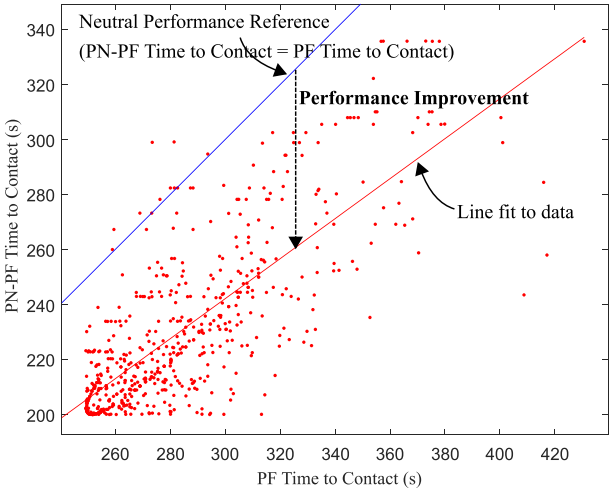

FIGURE 5. Results 1: Comparison of Time-to-Contact. Red data points below the blue line indicate performance improvement using PN-PF instead of PF alone. This plot shows 591 out of 600 of the red dots are below the blue line demonstrating a performance improvement in 98.5% of the trials.

Fig. 5 shows the results of the 1200 scenarios. The time-to-contact with the PF method verses the time-to-contact with

the proposed PN-PF method is plotted. A blue line with a slope of 1 is plotted as a performance reference. Any trial that results in both algorithms performing the same will land on this line. If the combined PN-PF guidance law developed in this paper performs better than PF alone then the trial will be plotted *below* the blue line. If the PN-PF guidance performs worse than the original PF guidance then the PN-PF trial will be plotted *above* the blue line. *These results show that 98.5% of trials exhibit better performance using the combined law method developed in this paper.*

To quantify the increase in performance of our PN-PF method, we took each data point and calculated the percent difference between it and results with PF alone, (25).

$$\text{PercentDiff}_{i,N} = 100 \cdot \frac{\left(Y_{i,N} - X_{i,N}\right)}{Y_{i,N}} \tag{25}$$

Where $Y_{i,N}$ is the data collected using just the potential field method (for the trial of density iteration N and interceptor location i), and $X_{i,N}$ is the data collected using the PN-PF method for this same trial. Plotting the data in a histogram shows that the combined algorithm gives an 18.8% increase in performance, Fig. 6.

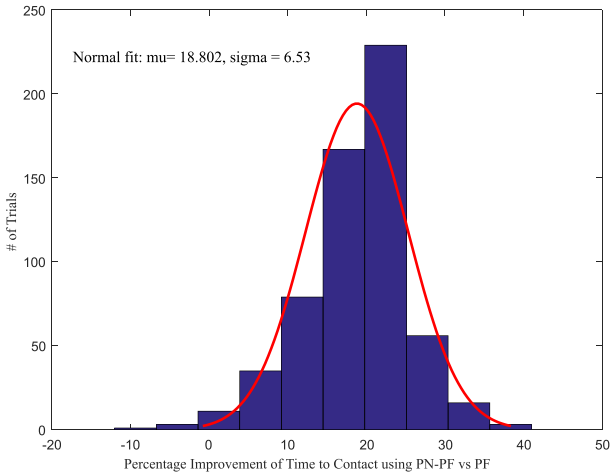

FIGURE 6. Results 1: Percent Difference of Algorithm Performances. This shows that on average the proposed PN-PF algorithm is approximately 18.8% faster than Potential Fields alone.

An analysis on the performance as a function of obstacle density was done for this data. All data from a particular obstacle density was averaged for both PN-PF and PF algorithms individually using (26) and (27) respectively (see Fig. 4 for the data tree guide). The results of these calculations allowed us to plot the mean time-to-contact as a function of the density of obstacles in the field, Fig. 7. As expected, as the density of the obstacles increases, the average time-to-contact also increases. The performance increase is still maintained around 18.8%.

$$\text{PF Avg Time-to-Contact}_N = \frac{1}{i_{max}} \cdot \sum_{i=1}^{i_{max}} Y_{i,N} \tag{26}$$

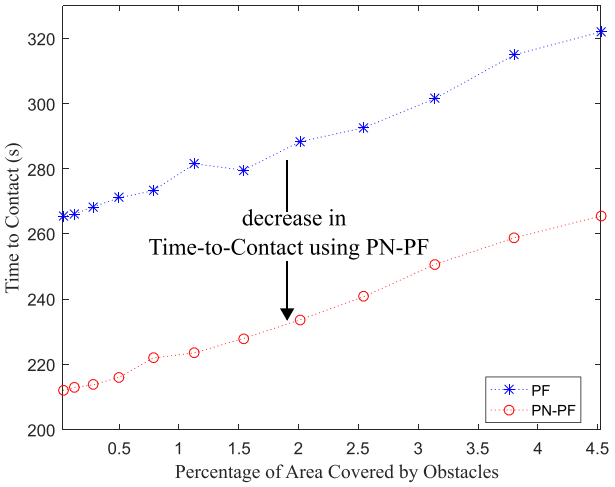

FIGURE 7. Results 1: Mean time-to-contact by density of uniformly placed obstacles. This chart shows performance for twelve different obstacle densities [N =1 to 12 points calculated using (26) and (27)]. Because the distance between the two lines remains roughly the same, this shows the Time-to-Contact advantage of the proposed PN-PF algorithm is robust to obstacle density when the obstacles are uniformly placed in the environment.

$$\text{PN-PF Avg Time-to-Contact}_N = \frac{1}{i_{max}} \cdot \sum_{i=1}^{i_{max}} X_{i,N} \tag{27}$$

Results 1 Analysis: The proposed PN-PF combined guidance algorithm developed in this paper combining Parallel Navigation and Potential Fields was shown to increase performance by reducing the time-to-contact. This increase in performance is maintained over different scenario geometries and obstacle densities.

B. RESULTS 2: RANDOMLY PLACED OBSTACLES AND DEFINED INTERCEPTOR STARTING LOCATIONS

Section IV-A demonstrated algorithm performance in the presence of uniformly distributed obstacles. This section's Matlab results provide performance results when the obstacles are *randomly placed* in the environment. Each obstacle's location is defined by an (x,y) center point which was drawn from a uniform distribution [0,200] for both the x and y coordinates. We placed a constraint on the random obstacles' creation so there is distance between each of them, and they do not overlap. The constraint is that obstacle center points are separated by a minimum distance. We chose a minimum distance that is a function of the obstacle's repulsive boundary specification, *RHO*. We set the minimum separation distance using, $(2 + RHO + 0.5 \cdot RHO)$, which in our simulations is 3.875 m. Fig. 1 shows an example of such a random obstacle environment for a density of 2.01%.

Fifty simulation iterations were performed for each of nine density scenarios. For example: for density iteration nine, fifty different obstacle scenarios were created where each had the same number of obstacles, but in different places. In each of these scenarios, the interceptor robot started at a different position. As shown in Fig. 3, the starting point of the

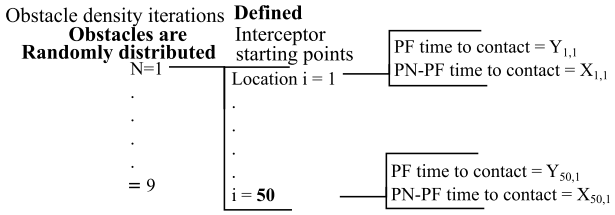

FIGURE 8. Results 2: Data tree for the Matlab experiments with randomly placed obstacles (which conform to a particular density for each N) and deterministic initial locations of the interceptor robot. In contrast to Fig 4, in these cases the obstacles were randomly placed in the environment (the interceptor robot still started at 50 pre-defined locations.)

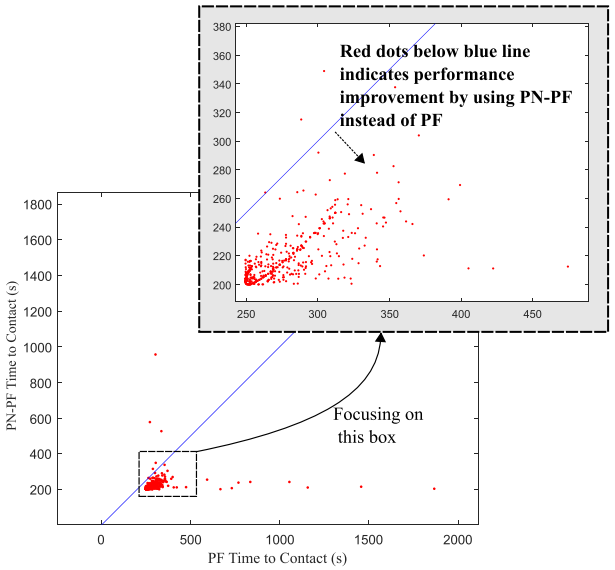

FIGURE 9. Results 2: Comparison of Time-to-Contact: Red data points below the blue line indicate performance improvement using PN-PF instead of PF alone. This plot shows 428 out of 434 of the red dots are below the blue line demonstrating a performance improvement in 98.6% of the trials.

interceptor robot was at one of fifty pre-determined positions along the x-axis between [0,100]. Fig. 8 shows the data tree description of this experiment. Fig. 9 shows the results of the data. The time-to-contact with the PF method verses the time-to-contact with the proposed PN-PF method is plotted. A red dot represents each simulation. A blue line with a slope of 1 is plotted as a performance reference. As before, red dots *below* the blue line indicate that our PN-PF algorithm is outperforming PF, and this data shows 98.6% of these red dots are below the blue line. The Fig. 9 data is consistent with the Fig. 5 data in that approximately 98% of the cases show our algorithm provides better (lower) time-to-contact performance than PF when the obstacles are uniformly or randomly spaced.

Similarly to Sec IV-A, an analysis on the performance as a function of obstacle density was done for this data. All data from a particular obstacle density was averaged for both PN-PF and PF algorithms individually using (26) and (27) respectively (see Fig. 8 for the data tree guide). The results of these calculations allowed us to plot the mean time-to-contact as a function of the density of obstacles in the field, Fig. 10.

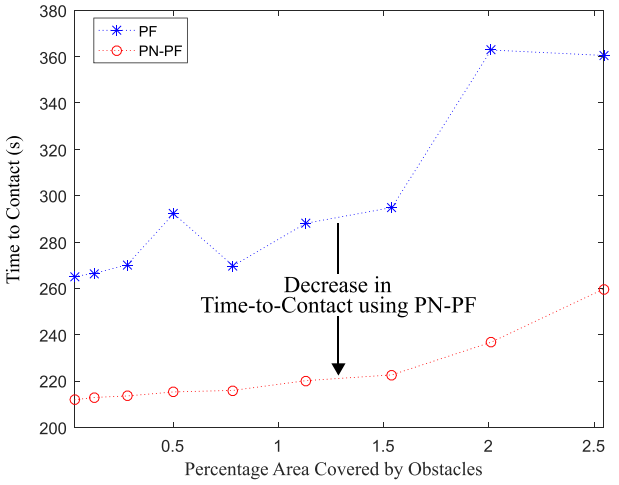

FIGURE 10. Results 2: Mean time-to-contact by density of randomly placed obstacles. This chart shows N=1 to 9 points calculated using (26) and (27). This shows that the average Time-to-Contact advantage of using the proposed PN-PF algorithm over PF alone is robust to obstacle density and Random placement of the obstacles. Observation: PN and PN-PF both had diminished performance for more dense and random environments. This is indicated because the contact times increase as obstacle percentage increases.

The data in Fig. 9 and Fig 10 represents 434 data points instead of the expected 450 (i.e. 9 densities multiplied by 50 data points per density equals 450). This is because cases which took longer than 4000 s were considered "No Solution" cases, and these 3.6% were removed from the analysis. Eight PF and one PN-PF cases with solutions traveled outside of the rectangular obstacle area before making contact. These are included in the statistics.

Results 2 Analysis: The PN-PF combined guidance algorithm's performance is robust even in more complex obstacle location scenarios than just uniform placement.

C. RESULTS 3: RANDOMLY PLACED OBSTACLES AND RANDOM INTERCEPTOR STARTING LOCATIONS

This section's results build upon Sec. IV-B by now adding randomness to the starting interceptor-target engagement geometry. Matlab simulations were conducted again with randomly placed obstacles as described in Sec IV-B. Fifty iterations were performed for each of nine obstacle density scenarios. The starting locations in these results, in contrast to the previous results, were drawn from a uniform distribution [0,100] for both the x and y coordinates. Fig. 11 shows the data tree description of this experiment. The time-to-contact with the PF method verses the time-to-contact with the proposed PN-PF method is plotted in Fig. 12. A red dot represents each simulation. A blue line with a slope of 1 is plotted as a performance reference. As before, red dots *below* the blue line indicate that our PN-PF algorithm is outperforming PF, and this data shows 96.1% of these red dots are below the blue line.

An analysis on the performance as a function of obstacle density was done for this data. All data from a particular obstacle density was averaged for both PN-PF and PF

TABLE 1. Summary of results.

Results Set	Obstacle placement	Interceptor Starting Location	Data Presented	Conclusions
1	Uniform	specified	Figs. 3 to 7	About 98% of the cases showed better performance using PN-PF over PF alone
2	Random	specified	Figs. 8 to 10	This showed that the algorithm is robust to different obstacle placements.
3	Random	Random	Figs. 11 to 13	This showed that the algorithm is robust to different interceptor-target starting engagement geometries.

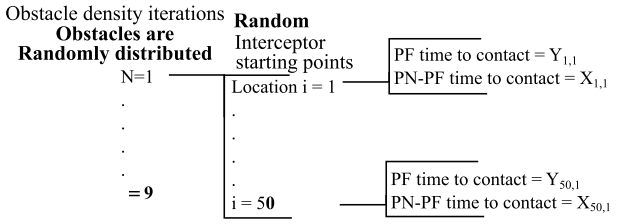

FIGURE 11. Results 3: Data tree for the Matlab experiments with Random obstacle locations and Random initial locations of the interceptor.

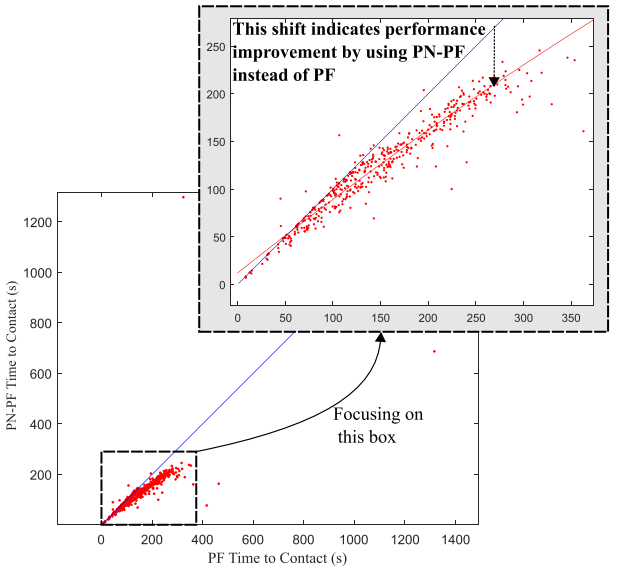

FIGURE 12. Results 3: Comparison of Time-to-Contact: Red data points below the blue line indicate performance improvement using PN-PF instead of PF alone. This plot shows 420 out of 437 of the red dots are below the blue line demonstrating a performance improvement in 96.1% of the trials.

algorithms individually using (26) and (27) respectively (see Fig. 8 for the data tree guide). The results of these calculations allowed us to plot the mean time-to-contact as a function of the density of obstacles in the field, Fig. 13.

The data in Fig. 12 and Fig 13 represents 437 data points instead of the expected 450 (i.e. 9 densities multiplied by 50 data points per density equals 450). This is because cases which took longer than 4000 s were considered ''No Solution'' cases, and these 2.9% were removed from the analysis. One PF and two PN-PF cases with solutions traveled outside of the rectangular obstacle area before making contact. These are included in the statistics.

Results 3 Analysis: The PN-PF combined guidance algorithm was shown to continue to increase performance by reducing the time-to-contact even in more complex obstacle location scenarios and over a variety of interceptor-to-target starting engagement geometries.

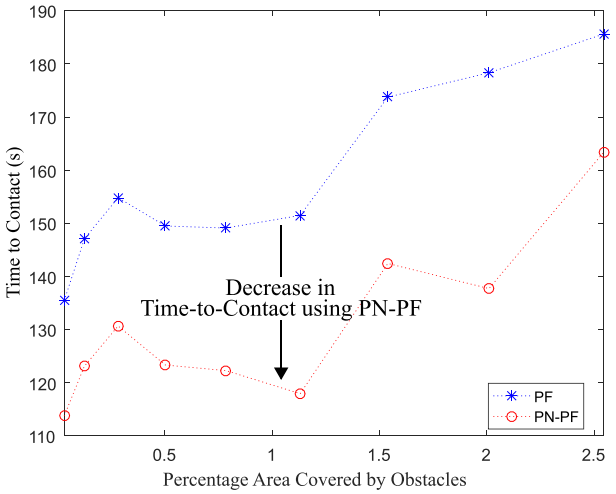

FIGURE 13. Results 3: Mean time-to-contact by density of randomly placed obstacles. This chart shows N=1 to 9 points calculated using (26) and (27). This shows that the average Time-to-Contact advantage of using the proposed PN-PF algorithm over PF alone is robust to obstacle density, Random placement of the obstacles, and random interceptor-target engagements. Observation: PN and PN-PF both had diminished performance for more dense and random environments.

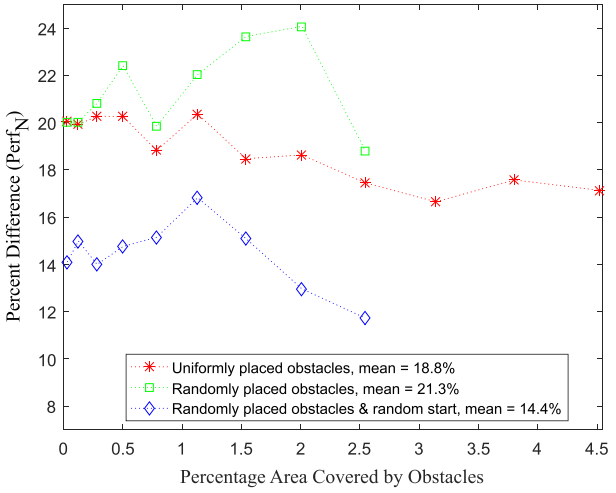

FIGURE 14. Results 1, 2 and 3: This data shows the average performance improvements for the three data sets. All the average percent differences are positive which indicates performance improvement by using PN-PF instead of PF. This shows that the algorithm is robust across different densities and interceptor-target engagement geometries.

V. CONCLUSION

The rendezvous guidance algorithm described in this paper combines aspects of both Parallel Navigation (PN) and Potential Fields (PF). It was tested against scenarios of varying obstacle densities and interceptor-target starting engagement geometries. It was found to provide a faster time-to-contact

in approximately 96-98% of the cases when compared to PF alone.

Fig. 14 summarizes and compares the three sets of performance results. We used a *percent difference* method to fairly compare these three sets of results which cover different densities and engagement geometries. The percent difference for each simulation in a density was calculated, and then the sum was divided by the number of valid cases, (28). The results are shown in Fig. 14 and demonstrate a 14.4-21.3% range of improvement.

$$\text{Perf}_N = \frac{100}{i_{countWithSolutions}} \cdot \sum_{i=1}^{i_{\max}=50} \frac{\left(Y_{i,N} - X_{i,N}\right)}{Y_{i,N}} \quad \forall i \in SolutionCases \tag{28}$$

Table 1 summarizes the results. Future studies can include adding dynamics and non-holonomic vehicle models to the simulations as well as sensor and navigation models. Three dimensional engagements can also be investigated [29], for example rendezvous for on-orbit satellite servicing [42].

REFERENCES

[1] R. Yanushevsky, *Guidance of Unmanned Aerial Vehicles*. Boca Raton, FL, USA: CRC Press, 2011.
[2] R. C. Michelson, "Official rules for the International Aerial Robotics Competition Mission 7," Version 13.0, SEPDAC, Inc., Canton, GA, USA, Jan. 2017. [Online]. Available: http://www.aerialroboticscompetition.org/assets/downloads/mission7rules_legacy.pdf
[3] I.-K. Jeong and J.-J. Lee, "Evolving fuzzy logic controllers for multiple mobile robots solving a continuous pursuit problem," in *Proc. IEEE Int. Fuzzy Syst. Conf. (FUZZ-IEEE)*, vol. 2. Aug. 1999, pp. 685–690.
[4] F. Belkhouche, B. Belkhouche, and P. Rastgoufard, "Line of sight robot navigation toward a moving goal," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 2, pp. 255–267, Apr. 2006.
[5] J. H. Choi, W.-S. Lee, and H. Bang, "Helicopter guidance for vision-based tracking and landing on a moving ground target," in *Proc. IEEE 11th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2011, pp. 867–872.
[6] G. B. Palmerini, M. Sabatini, A. Pisculli, and P. Gasbarri, "Ground tests of a rendezvous maneuver based on visual servoing," in *Proc. IEEE Aerosp. Conf.*, Mar. 2014, pp. 1–14.
[7] F. Tyan, "Capture region of a 3D PPN guidance law for intercepting high-speed targets," *Asian J. Control*, vol. 14, no. 5, pp. 1215–1226, 2012.
[8] S. Ghosh, D. Ghose, and S. Raha, "Capturability analysis of a 3-D Retro-PN guidance law for higher speed nonmaneuvering targets," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 5, pp. 1864–1874, Sep. 2014.
[9] H. M. Prasanna and D. Ghose, "Retro-proportional-navigation: A new guidance law for interception of high speed targets," *J. Guid., Control, Dyn.*, vol. 35, no. 2, pp. 377–386, 2012.
[10] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib, "Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 3, pp. 238–250, May 2000.
[11] D. Chwa, J. Kang, and J.-Y. Choi, "Online trajectory planning of robot arms for interception of fast maneuvering object under torque and velocity constraints," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 6, pp. 831–843, Nov. 2005.
[12] A. A. Masoud and A. Al-Shaikhi, "Time-sensitive, sensor-based, joint planning and control of mobile robots in cluttered spaces: A harmonic potential approach," in *Proc. 54th IEEE Conf. Decision Control (CDC)*, Dec. 2015, pp. 2761–2766.
[13] A. A. Masoud, "Motion planning with gamma-harmonic potential fields," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 2786–2801, Oct. 2012.
[14] L. Huang, "Velocity planning for a mobile robot to track a moving target—A potential field approach," *Robot. Auto. Syst.*, vol. 57, no. 1, pp. 55–63, 2009.
[15] I. Lopez and C. R. McInnes, "Autonomous rendezvous using artificial potential function guidance," *J. Guid., Control, Dyn.*, vol. 18, no. 2, pp. 237–241, 1995.
[16] H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 541–552, Apr. 2015.
[17] C. Heller and I. Yaesh, "Proportional navigation with integral action," in *Proc. 15th IEEE Medit. Electrotech. Conf. (MELECON)*, Apr. 2010, pp. 1546–1550.
[18] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[19] G.-H. Li, C.-F. Chang, and L.-C. Fu, "Navigation of a wheeled mobile robot in indoor environment by potential field based-fuzzy logic method," in *Proc. IEEE Workshop Adv. Robot. Social Impacts*, Aug. 2008, pp. 1–6.
[20] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proc. Congr. Evol. Comput.*, vol. 1. Jul. 2000, pp. 256–263.
[21] V. Gazi, B. Fidan, R. Ordóñez, and M. İ. Köksal, "A target tracking approach for nonholonomic agents based on artificial potentials and sliding mode control," *J. Dyn. Syst., Meas., Control*, vol. 134, no. 6, p. 061004, 2012.
[22] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2347–2354.
[23] M. G. Slack, "Navigation templates: Mediating qualitative guidance and quantitative control in mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 2, pp. 452–466, Mar. 1993.
[24] S.-C. Han and H. Bang, "Proportional navigation-based optimal collision avoidance for UAVs," in *Proc. 2nd Int. Conf. Auto. Robots Agents*, 2004, pp. 13–15.
[25] F. Kunwar, F. Wong, R. B. Mrad, and B. Benhabib, "Time-optimal rendezvous with moving objects in dynamic cluttered environments using a guidance based technique," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 283–288.
[26] F. Kunwar, F. Wong, R. B. Mrad, and B. Benhabib, "Rendezvous guidance for the autonomous interception of moving objects in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 3776–3781.
[27] F. Kunwar and B. Benhabib, "Rendezvous-guidance trajectory planning for robotic dynamic obstacle avoidance and interception," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1432–1441, Dec. 2006.
[28] F. Kunwar, P. K. Sheridan, and B. Benhabib, "Predictive guidance-based navigation for mobile robots: A novel strategy for target interception on realistic terrains," *J. Intell. Robot. Syst.*, vol. 59, nos. 3–4, pp. 367–398, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10846-010-9401-3
[29] P. L. Friudenberg, "Integration of potential field theory and proportional navigation theory to autonomously guide an unmanned aerial vehicle," M.S. thesis, Dept. Elect. Comput. Eng., Baylor Univ., Waco, TX, USA, 2015.
[30] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Basic principles of homing guidance," *Johns Hopkins APL Tech. Dig.*, vol. 29, no. 1, pp. 25–41, 2010.
[31] H. Chen, K. Chang, and C. S. Agate, "UAV path planning with tangent-plus-Lyapunov vector field guidance and obstacle avoidance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 840–856, APR. 2013.
[32] S. Zhu, D. Wang, and Q. Chen, "Standoff tracking control of moving target in unknown wind," in *Proc. 48th IEEE Conf. Decision Control, 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 776–781.
[33] M. Fraiwan, A. Alsaleem, H. Abandeh, and O. Aljarrah, "Obstacle avoidance and navigation in robotic systems: A land and aerial robots study," in *Proc. IEEE 5th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2014, pp. 1–5.
[34] J.-W. Choi, "A potential field and bug compound navigation algorithm for nonholonomic wheeled robots," in *Proc. IEEE 1st Int. Conf. Innov. Eng. Syst. (ICIES)*, Dec. 2012, pp. 166–171.
[35] Z.-Y. Liu, R.-H. Jing, X.-Q. Ding, and J.-H. Li, "Trajectory tracking control of wheeled mobile robots based on the artificial potential field," in *Proc. IEEE 4th Int. Conf. Natural Comput.*, vol. 7. Oct. 2008, pp. 382–387.
[36] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2. Mar. 1985, pp. 500–505.

[37] S. Lim and H. Bang, "Guidance laws for target localization using vector field approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 1991–2003, Jul. 2014.
[38] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for small unmanned air vehicles," in *Proc. IEEE Amer. Control Conf.*, Jun. 2006, pp. 1–7.
[39] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 519–529, Jun. 2007.
[40] S. A. Murtaugh and H. E. Criel, "Fundamentals of proportional navigation," *IEEE Spectr.*, vol. 3, no. 12, pp. 75–85, Dec. 1966.
[41] F. N. Martins, M. Sarcinelli-Filho, and R. Carelli, "A velocity-based dynamic model and its properties for differential drive mobile robots," *J. Intell. Robot. Syst.*, vol. 85, no. 2, pp. 277–292, Feb. 2017. [Online]. Available: https://doi.org/10.1007/s10846-016-0381-9
[42] X. Chen and S. Qin, "Kinematic modeling for a class of free-floating space robots," *IEEE Access*, vol. 5, pp. 12389–12403, 2017.

PATRICK FRIUDENBERG received the B.S.E.C.E. and M.S.E.C.E. degrees in electrical and computer engineering from Baylor University, Waco, TX, USA, in 2015. He is currently a member of HKN.

SCOTT KOZIOL (M'05) received the B.S.E.E. degree in electrical engineering from Cedarville University, Cedarville, OH, USA, in 1998, the M.S. degree in electrical engineering from Iowa State University, Ames, in 2000, and the M.S.M.E. degree in mechanical engineering and the Ph.D. degree in robotics from the Georgia Institute of Technology, Atlanta, in 2011 and 2013, respectively. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX, USA.