Advanced Predictive Guidance Navigation for Mobile Robots: A Novel Strategy for Rendezvous in Dynamic Settings

Faraz Kunwar and Beno Benhabib

Computer Integrated Manufacturing Laboratory
Department of Mechanical and Industrial Engineering, University of Toronto
5 King's College Road, Toronto, Ontario, Canada, M5S 3G8
email: kfaraz@mie.utoronto.ca

***Abstract*—**This paper presents a novel on-line trajectory planning method for the autonomous robotic interception of moving targets in the presence of dynamic obstacles, i.e., position and velocity matching (also referred to as *rendezvous*). The novelty of the proposed time-optimal interception method is that it directly considers the dynamics of the obstacles as well as the target in its interception maneuver: the velocities and accelerations of the obstacles and the target are predicted in real-time for potential collisions. The method is designed to deal with highly-maneuvering obstacles and targets. The interception maneuver is computed using an Advanced Predictive Guidance Law.

Extensive simulation and experimental analyses, some of which are reported in this paper, have clearly demonstrated the time efficiency of the proposed rendezvous method.

***Index Terms*—** Target interception, on-line trajectory planning, rendezvous guidance.

I. INTRODUCTION

Real world mobile robotic environments have, typically, time-varying topologies, with numerous *objects* moving with respect to each other. Furthermore, such environments are subject to uncertainties as complete information and future trajectories of objects cannot be assumed to be known *a priori*. Thus, there rises a need for autonomous routing decisions – on-line motion planning and execution of robotic *vehicle* trajectories. A preferred solution to this problem would be one that takes into consideration the kinematic constraints of the vehicle, explicitly copes with dynamically moving objects, and is analytical. Furthermore, high-level autonomy and time optimality would be desirable during motion planning via real-time sensory-data collection about the objects. In this context, the focus of this paper is the development of a generic guidance-based methodology that would provide an autonomous robotic vehicle with the capability to time-optimally *rendezvous* with a moving target (matching position and velocity) in the presence of mobile obstacles.

The problem of robotic-vehicle interception in obstacle-cluttered environments using a Rendezvous-Guidance (RG) method augmented with a Modified Exact Cell Decomposition method for rendezvous was first addressed in [1]. The time optimality of this method was further improved in [2] through the use of a Velocity Obstacle (VO) approach. The modified method, however, still only ensured near-optimal performance for rendezvous with non-maneuvering targets. Therefore, herein, a new guidance law that can yield optimal rendezvous with targets having high degree of maneuverability is proposed: This novel guidance law utilizes an Extended Kalman Filter (EKF) to estimate the target's state, as well as directly makes use of the dynamic characteristics of the obstacles to estimate their future states. These estimates are used to predict potential collisions between the pursuer and the obstacles by using the *Collision Cone* (CC) method. A brief overview of the pertinent literature is provided below.

A. Guidance-Based Interception

Missile-guidance techniques have been classified into five main categories: Line-Of-Sight (LOS) guidance; Pure Pursuit (PP); Proportional Navigation Guidance (PNG); Optimal Guidance (OG); and, other guidance methods including the use of differential game theory [3, 4]. Missile-guidance laws assume that the future trajectory of the target is completely defined either analytically or by a probabilistic model [5-7]. However, the problem of velocity matching, as

dealt in this paper, is not typically addressed in missile-guidance applications.

The PNG law uses the homing triangle for computing the acceleration of an interceptor pursuing an evading target. The homing triangle is defined by the pursuer, the target, and the point of interception. This control law makes the pursuer's acceleration normal to its path and proportional to the rate of change of the LOS vector to the target. Due to its low computational requirements, simplicity of on-board implementation, and time optimality characteristics, PNG has been the most widely used guidance technique [8]. The need for velocity matching has resulted in a new class of guidance methods, commonly referred to as Rendezvous-Guidance (RG) methods. A PNG-based RG method for the docking problem of two space vehicles was proposed in [9]. In [10], the use of exponential-type guidance was suggested for asteroid rendezvous. The problem of rendezvous with an object capable of performing evasive maneuvers in order to avoid rendezvous was addressed in [11].

The utilization of a guidance-based technique in robot-motion planning, with the purpose of improving upon the interception time achievable by visual-servoing techniques, was first reported in [12-14]. Although, these works showed that guidance-based methods could yield shorter interception times compared to other available techniques, all were limited to environments with no obstacles. Further work carried out in our laboratory augmented guidance-based methods with the capability to avoid mobile obstacles [1-2, 15]. These are based on the PN technique, which basically deals with first-order derivatives of target/obstacle velocity, though, they are not very effective in the presence of highly maneuvering target/obstacles.

In present day applications, the maneuverability of the target as well as the obstacles is increasingly becoming more complex. Especially in cases that involve human-machine interactions, for example, in public places like museums, hospitals, or factory floors. Such applications necessitate better tracking of the obstacles as well as of the target than is achievable by models based on PN techniques. The reason for the inadequate tracking performance of such models is that the higher-order derivatives in the case of very highly maneuvering targets are significant. Therefore, in this paper, we present a novel method that can deal with highly-maneuvering targets by inherently considering the higher-order terms of the target/obstacle model.

B. Obstacle Avoidance

Motion-planning problems for mobile robots have been classified as *static* or *dynamic*. For the former, the obstacle information is assumed to be known to the planner in its entirety prior to planning. For the latter, information about the environment becomes known to the planner only during run-time and often during the execution of a partially-constructed plan.

Static motion-planning approaches, such as potential field and vector field histogram, calculate the desired motion direction and steering commands in two separate steps [16-18]. In the first step, the obstacle-avoidance method provides intermediate destination points that connect a collision-free path from the robot to the target. In the second step, acceleration commands are derived for the path generated for the motion of the robot. Such a methodology would not be acceptable for a dynamic environment with fast moving obstacles, where the uncertainty about the environment prevents the computation of a solution that is guaranteed to succeed. Furthermore, static obstacle-avoidance methods have been developed to deal with geometric constraints, more specifically, *holonomic* systems. For non-holonomic systems such as mobile robots, kinematic constraints make time derivatives of some configuration variables non-integrable and, hence, a collision-free path in the configuration space is not necessarily feasible (i.e., it may not be achievable by steering controls) [19-20].

In dynamic obstacle avoidance methods, information about the environment becomes known to the planner only during runtime and often during the execution of a partially constructed plan. The Curvature-Velocity (CV) [21] and the Dynamic Window (DW) [22] methods are based on the *steer-angle-field* approach [23]. The CV method chooses a location in translational and rotational velocity space which satisfies constraints placed on the robot and maximizes an objective function [24]. The Lane Curvature Method [25] improves upon the CV method by using a directional-lane method. The DW method considers the kinematic and dynamic constraints of a mobile robot [26]. Kinematic constraints are taken into account by directly searching the velocity space of the robot. The search space is reduced to a dynamic window representing the velocities achievable by the robot in a given interval of time. In spite of the good results for obstacle avoidance at high velocities achieved by both CV and DW methods, local-minima problem persists. In order to overcome this shortcoming, the DW Method was integrated with a gross-motion planner in [27] and extended to use a map in conjunction with sensory information in [28] to generate collision-free motions.

The above approaches require *a priori* information about the environment. The Velocity Obstacle (VO) approach proposed in [29], on the other hand, determines potential collisions and computes collision-free paths for robots moving in dynamic environments. The VO method was extended in [30] to include objects moving along non-linear trajectories.

The work presented in this paper uses concepts that are rooted in missile guidance/aerospace literature for obstacle avoidance so that the method provides an elegant integration with the navigation algorithm used for interception. Our objective is, thus, developing a novel obstacle-avoidance navigation law based on proven navigation-guidance principles.

Up to now, most of the existing methods have dealt with obstacle avoidance by non-holonomic systems in one of two ways. The first is to exclusively focus upon motion planning under non-holonomic constraints without considering obstacles – differential geometry [31], differential flatness [32], input parameterization [33-35], and optimal control [36]. In particular, the non-holonomic motion-planning problem is recast as an optimal control problem, where Pontryagin's Maximum Principle is applied. As first shown in [37] and later improved in [38], the feasible shortest path for a point robot under two boundary conditions is a concatenation of simple pieces (such as an arc and a straight line segment) that belong to three-parameter families of controls.

The second way in dealing with obstacle avoidance by non-holonomic systems is to modify the resultant solution from a holonomic planner so that the resulting path is feasible. For example, the online sub-optimal obstacle avoidance algorithm in [39] is based on the Hamilton–Jacobi–Bellman equation ([40], [41]), deals only with stationary obstacles, and the planned path is holonomic, whose feasibility has to be verified in case of a non-holonomic mobile robot. Similarly, the non-holonomic path planner in [42] generates a path by ignoring non-holonomic constraints and it is, then, made feasible via approximation by using a sequence of optimal path segments such as those in [43].

In this paper, we consider the non-holonomic constraints of the pursuer directly within the navigation algorithm. It is ensured that all acceleration commands generated by the navigation/obstacle-avoidance algorithm are within the acceleration capabilities of the pursuer and can be executed at the next time instant. Although the algorithm presented in this paper is primarily designed for vehicular robots yet it can also be applied to humanoid robots, e.g., the

control system for a biped presented in [44] can easily be integrated with the proposed algorithm to enable the robot to avoid static and dynamic obstacles while moving in the desired direction.

II. PROPOSED SYSTEM

A schematic diagram of the proposed rendezvous system is shown in Figure 1: First, the states of the pursuer, obstacles, and target are estimated and sent to the path planner; The path planner, then, generates a single acceleration command for the pursuer, for time-optimal rendezvous while avoiding obstacles (as detailed in Section III) – if obstacle avoidance is not needed, the pursuer obtains its acceleration command directly from the navigation algorithm.

In the following sub-sections, first the methodology for obtaining a model of the pursuer is detailed and, subsequently, the proposed novel guidance and obstacle-avoidance methods are presented.

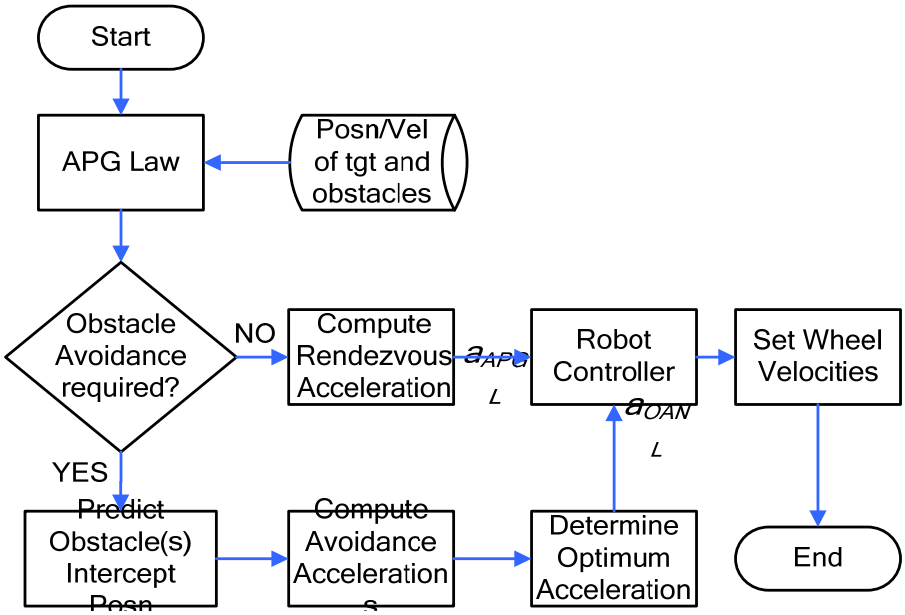


Figure 1: Proposed Rendezvous System.

A. The Kinematic Model of the Mobile Robot

The kinematic model for a differentially-driven wheeled mobile robot (as the ones used in our experiments) is selected as the basis for this work:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} \cos\lambda & 0 \\ \sin\lambda & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}, \qquad \begin{aligned} y &= \begin{bmatrix} x_c & y_c & \lambda \end{bmatrix}^T \\ u &= \begin{bmatrix} \boldsymbol{v} & \boldsymbol{\omega} \end{bmatrix}^T \end{aligned}, \tag{1}$$

where y is the system state; the robot is located at (x_c, y_c) turning to the right; λ is the robot heading angle with respect to the X-axis; and, the control u consists of the linear velocity $\boldsymbol{v}$ and the angular velocity $\boldsymbol{\omega}$. Although in (1) the controls of the mobile robot are its linear and angular velocities, the actual commands provided to the vehicle are the right and left wheel velocities, Figure 2.

Let $\boldsymbol{v_l}$, $\boldsymbol{v_r}$, and $\boldsymbol{v_R}$ represent the velocities of the left wheel, the right wheel, and the robot, respectively. Also, let d be the distance between the two wheels and D be the distance between the right wheel and the Instantaneous Center of Curvature, *ICC*. The commands generated by the navigation-guidance/obstacle-avoidance algorithm set the linear velocity, $\boldsymbol{v_P}$, and the angular velocity, $\boldsymbol{\omega_P}$, of the pursuer:

$$\boldsymbol{\omega_P} = \frac{\boldsymbol{v_P}}{d/2 + D}. \tag{2}$$

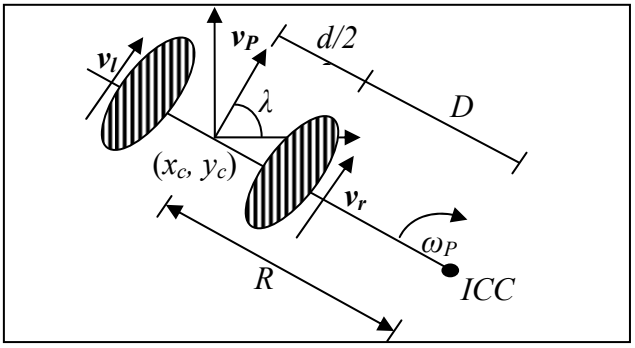


Figure 2: Wheel Velocities.

The motion commands are executed by specifying $\boldsymbol{v_l}$, and $\boldsymbol{v_r}$. With d known, it is possible to determine R, the turning radius of curvature of the robot, as the distance between the centre of

the robot and *ICC*. The wheel velocities are determined using the kinematics Equations (3) to (5) given below,

$$\boldsymbol{v_P} = \frac{\boldsymbol{v_l} + \boldsymbol{v_r}}{2}, \tag{3}$$

$$\frac{\boldsymbol{v_r}}{D} = \frac{\boldsymbol{v_l}}{D+d}, \quad \text{and} \tag{4}$$

$$\boldsymbol{v_l} = r_{wh}\boldsymbol{\omega_l}, \boldsymbol{v_r} = r_{wh}\boldsymbol{\omega_r}, \tag{5}$$

where r_{wh} is the radius of the wheel, and $\boldsymbol{\omega_l}$ and $\boldsymbol{\omega_r}$ are the angular velocities of the left and right wheel, respectively.

The final velocity for the next time instant, as computed by the proposed algorithm, must be one that is achievable by the robot. Therefore, we define herein a *Feasible Velocity Region* (*FVR*) representing all the velocities achievable by the robot within Δt, taking into account the kinematic and dynamic constraints on the robot [2], Figure 3:

$$FV(t_i + \Delta t) = \left\{ \boldsymbol{v} \mid \boldsymbol{v} = \boldsymbol{v_P}(t_i) \oplus \Delta t \bullet FA(t_i) \right\}, \tag{6}$$

where $FA(t_i)$ represents the set of feasible accelerations of the robot at time t_i. The *FVR* polygon is computed by scaling $FA(t_i)$ by Δt and adding it to the current velocity of the pursuer $\boldsymbol{v_P}$. Thus, by ensuring that the algorithm only generates accelerations that are within the *FVR* for each time instant, we ensure that the non-holonomic constraints of the mobile robot are satisfied.

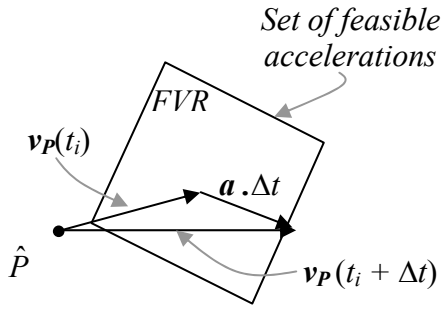


Figure 3: Feasible Accelerations.

B. Interception Using the Advanced Predictive Guidance Law

This section describes the new Advanced Predictive Guidance Law (*APGL*) based target-interception method proposed in this paper. The Proportional Navigation Law (*PNL*) is reviewed first to provide the mathematical basis for the *APGL*.

Theoretically, the *PNL* issues commands perpendicular to the instantaneous pursuer-target line-of-sight, which are proportional to the line of sight rate and closing velocity:

$$\boldsymbol{n}_c\big|_{PNL} = N\boldsymbol{v}_c\dot{\lambda}\,, \tag{7}$$

where $\boldsymbol{n}_c$ is the acceleration command, N is a unit-less gain (usually in the range of 3-5) known as the *effective navigation ratio* [3], $\boldsymbol{v}_c$ is the pursuer-target closing velocity, and λ is the *LOS* angle or the *LOS* rate. The two-dimensional, point-mass target-engagement geometry for *PNL* is shown in Figure 4: the pursuer with the velocity $\boldsymbol{v}_P$, the pursuer is heading at an angle of (L+*HE*) with respect to the *LOS*; the angle L is known as the pursuer *Lead Angle*, which is the theoretically correct angle for the pursuer to be on a collision triangle with the target; the angle *HE* is known as the *Heading Error*, which represents the initial deviation of the pursuer from the collision triangle; and, the imaginary line connecting the pursuer and the target is known as the *LOS*. The *LOS* makes an angle λ with respect to the fixed reference frame and the length of the *LOS* (instantaneous separation between pursuer and target) is a range denoted by R_{TP}.

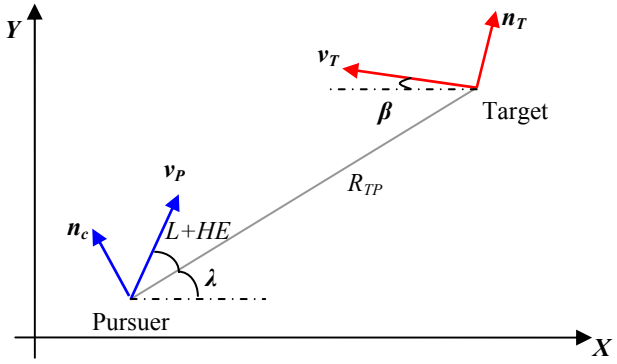


Figure 4: Two-dimensional Pursuer-Target Engagement Geometry.

From a guidance point of view, it is desirable that R_{TP}, at the expected time of intercept, is as small as possible (ideally, zero). The point of closest approach of the pursuer and target is known

as the *miss distance*. The closing velocity, $\boldsymbol{v_c}$, is defined as the negative rate of change of the distance from the pursuer to the target:

$$\boldsymbol{v_c} = -\dot{R}_{TP}\,. \tag{8}$$

Therefore, at the end of the interception, when the pursuer and the target are in closest proximity, the sign of $\boldsymbol{v_c}$ would change. The desired acceleration command $\boldsymbol{n_c}$ is perpendicular to the instantaneous line of sight.

As shown in Figure 4, the target can maneuver evasively with an acceleration $\boldsymbol{n_T}$. The angular velocity of the target is, thus, expressed as:

$$\dot{\beta} = \frac{\boldsymbol{n_T}}{\boldsymbol{v_T}}, \tag{9}$$

where $\boldsymbol{v_T}$ is the is the magnitude of the target velocity. Since the acceleration command is perpendicular to the instantaneous *LOS*, the pursuer acceleration command in the fixed reference frame has the following components:

$$\begin{aligned} \boldsymbol{a_{Px}} &= -\boldsymbol{n_c}\big|_{PNL} \sin\lambda \\ \boldsymbol{a_{Py}} &= \boldsymbol{n_c}\big|_{PNL} \cos\lambda \end{aligned}\,. \tag{10}$$

A pursuer employing *PNL* does not move directly toward the target, but, in a direction to *lead* the target. The theoretical pursuer *Lead* angle, *L*, can be found by application of the *law of sines*, yielding:

$$L = \sin^{-1}\frac{\boldsymbol{v_T}\sin(\beta+\lambda)}{\boldsymbol{v_P}}\,. \tag{11}$$

The *LOS* angle is, then, expressed in terms of the relative separation components:

$$\lambda = \tan^{-1}\frac{R_{TPy}}{\boldsymbol{v_{TPx}}}\,. \tag{12}$$

The *LOS* rate is expressed:

$$\dot{\lambda} = \frac{R_{TPx}\boldsymbol{v}_{\boldsymbol{TPy}} - R_{TPy}\boldsymbol{v}_{\boldsymbol{TPx}}}{R_{TP}^{\ 2}} \tag{13}$$

Any initial angular deviation of the pursuer from the collision triangle is defined by the angle *HE*. The initial pursuer velocity components are, therefore, expressed in terms of the theoretical lead angle, *L*, and actual heading error, *HE*, as

$$\begin{aligned} \boldsymbol{v}_{\boldsymbol{Px}}(0) &= \boldsymbol{v}_{\boldsymbol{P}} \cos(L + HE + \lambda) \\ \boldsymbol{v}_{\boldsymbol{Py}}(0) &= \boldsymbol{v}_{\boldsymbol{P}} \sin(L + HE + \lambda) \end{aligned} \tag{14}$$

Let us define a *Zero-Effort Miss* (*ZEM*, the bracketed term in Equation (15) below) as a prediction of by how much the pursuer would miss the target, if the target were to continue as it has done in the past and the pursuer was issued no further acceleration commands (i.e., *zero effort*). As shown in [3], *PNL* can also be considered as a guidance law in which the acceleration command is proportional to the *ZEM* and inversely proportional to the square of the time remaining to intercept:

$$\boldsymbol{n}_{\boldsymbol{c}}\big|_{PNL} = \frac{N'}{t_{go}^2}[y + \dot{y}t_{go}], \tag{15}$$

where y is the relative distance between the pursuer and the target, $\dot{y}$ is the relative target rate and t_{go} is the *time to go* before the intercept occurs ($t_{go} = R_{TP}/\boldsymbol{v}_{\boldsymbol{c}}$). Thus, for *PNL, ZEM* assumes that the target is not maneuvering. This does not imply that the *PNL* cannot intercept maneuvering targets, but rather that it is not optimal in their interception.

Advanced Predictive Guidance Law

In order to overcome the limitations of *PNL*, an Advanced Predictive Guidance Law, *APGL*, is proposed in this paper to improve the performance of a pursuer for fast-maneuvering targets. In this guidance law, the predicted intercept with the target is calculated on-line by integrating the non-linear pursuer and target equations forward in time at each guidance update. This proposed law overcomes the limitations of PN to improve the performance of an interceptor especially against a maneuvering target. As with PN, the APG tries to yield zero miss distance while minimizing the following:

$$y(t_F) = 0 \quad \text{subject to minimizing} \quad \int_0^{t_F} \boldsymbol{a}_c^2(t)dt. \tag{16}$$

The proposed APG law can be expressed in state-space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dddot{y}_T \\ \ddddot{y}_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix}}_{F} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y}_T \\ \dddot{y}_T \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}}_{G} \boldsymbol{a}_c, \tag{17}$$

where ω is the target maneuver frequency, y is the relative position between the pursuer and the target, $\dot{\boldsymbol{y}}$ is the relative velocity between the pursuer and the target, $\ddot{\boldsymbol{y}}_{\boldsymbol{T}}$ is the target acceleration and $\dddot{\boldsymbol{y}}_{\boldsymbol{T}}$ is the target jerk. The final state of the system at any time can also be expressed as

$$x(t_F) = \Phi(t_F - t)x(t) + \int_t^{t_F} \Phi(t_F - \lambda)G(\lambda)u(\lambda)d(\lambda), \tag{18}$$

where $x(t)$ is the system state vector and $\Phi(t)$ is the fundamental matrix and is related to F according to

$$\Phi(t) = L^{-1}\left[(sI - F)^{-1}\right]. \tag{19}$$

Solving (19) yields

$$\Phi(t) = \begin{bmatrix} 1 & t & \dfrac{(1-\cos\omega t)}{\omega^2} & \dfrac{(1-\cos\omega t)}{\omega^3} \\ 0 & 1 & \dfrac{\sin\omega t}{\omega} & \dfrac{(1-\cos\omega t)}{\omega^2} \\ 0 & 0 & \cos\omega t & \dfrac{\sin\omega t}{\omega} \\ 0 & 0 & -\omega\sin\omega t & \cos\omega t \end{bmatrix}, \tag{20}$$

Substituting Φ and G matrices into (18) and simplifying yields the proposed *APGL* for maneuvering targets which is expressed as

$$\boldsymbol{a}_c\big|_{APG} = 3\boldsymbol{v}_c\dot{\boldsymbol{\lambda}} + \frac{3}{t_{go}^{\,2}}\left[\frac{1-\cos\omega t_{go}}{\omega^2}\right]\ddot{\boldsymbol{y}}_{\boldsymbol{T}} + \frac{3}{t_{go}^{\,2}}\left[\frac{\omega t_{go} - \sin\omega t_{go}}{\omega^3}\right]\dddot{\boldsymbol{y}}_{\boldsymbol{T}}. \tag{21}$$

The above Equation (21) shows that in essence the APG is similar to the PN and APN wherein, the guidance commands are still proportional to the *ZEM* and inversely proportional to the square of t_{go}. However, the new guidance law consists of three terms: one proportional to the *LOS* rate, another proportional to the target acceleration, and a third proportional to the target jerk.

In order to better understand the relationship between the new guidance law and its predecessors, let us consider the case in which the target is not maneuvering, i.e., the target maneuvering frequency is zero. By using Taylor series approximation, the APG law is simplified to:

$$\lim_{\omega \to 0} \boldsymbol{a}_c \big|_{APG} = \frac{3}{t_{go}^2}\left[y + y t_{go} + \frac{t_{go}^2}{2}\ddot{\boldsymbol{y}}_{\boldsymbol{T}} + \frac{t_{go}^3}{6}\dddot{\boldsymbol{y}}_{\boldsymbol{T}} \right]. \tag{22}$$

The above is simply an Augmented Proportional Navigation Law (*APNL*) with an effective navigation ratio of 3 plus an extra term to account for target jerk, which implies that *APGL* requires, in addition to the *LOS* rate, an estimate of the target maneuver frequency, target jerk, and t_{go}. These terms can be estimated in a number of ways for example, by using external sensors or by implementing an Extended Kalman Filter (EKF). In this thesis, a five-state EKF is used to predict target characteristics. Details on how to implement a suitable EKF for this problem are given in [7]. A general method for the development of an EKF is given in Appendix A.

Thus, after the estimation all the information necessary to calculate $\boldsymbol{a}_c \big|_{APG}$, using (21), is now available. The acceleration for the next time instant using *APGL* is, finally, obtained by using the value obtained from (21) in (10).

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{APGx}} &= -\boldsymbol{a}_c \big|_{APG} \sin\lambda, \text{ and} \\ \boldsymbol{a}_{\boldsymbol{APGy}} &= \boldsymbol{a}_c \big|_{APG} \cos\lambda. \end{aligned} \tag{23}$$

Generating the Rendezvous Command

If the pursuer were to follow the acceleration commands generated in (23), it would intercept the target at an optimal time in the future. However, in order to rendezvous with the target, the velocity of the pursuer must also match the velocity of the maneuvering target at the time of interception.

Let us assume that the deceleration capability of the robot in the direction of motion is given by A. This acceleration would be used to bring the closing velocity down to zero. In deciding on the rendezvous maneuver, there are two primary issues that need to be resolved: First, the magnitude of the maximum closing velocity needs to be determined and, second, the time instant to switch between target-interception and target-rendezvous strategies needs to be carefully chosen (the former would lead to a collision with the target).

Let us denote $\dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}}$ as the magnitude of the maximum allowable closing/rendezvous velocity (hence, the superscript *rend*), t_{go} as the time remaining to intercept the target from the current instant, and R_{TP} as the relative distance between the pursuer and the target. In order to simultaneously reduce the relative velocity and the relative distance to zero, the following expressions need to be implemented:

$$\begin{aligned} &\boldsymbol{v}(t) = \dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}} - \boldsymbol{A}t_{go} \\ &\dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}} - \boldsymbol{A}t_{go} = 0 \end{aligned}, \text{ and} \tag{24}$$

$$\begin{aligned} &r(t) = R_{TP} - \int_0^{t_F} \boldsymbol{v}(\tau)d\tau \\ &R_{TP} - \dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}} t_{go} - \frac{1}{2}\boldsymbol{A}t_{go}^2 = 0 \end{aligned}. \tag{25}$$

The maximum instantaneous allowable closing velocity is obtained by solving (24) and (25):

$$\dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}} = \sqrt{2\boldsymbol{r}A}\,. \tag{26}$$

The maximum closing velocity as imposed by the frequency of velocity command generation by the trajectory planner for a fast asymptotic interception is given by:

$$\dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{cr}} = \boldsymbol{r}/_{n\Delta t}\,. \tag{27}$$

The value of n above is determined experimentally. The final allowable closing velocity component of the velocity command is, then, obtained by considering (26) and (27) simultaneously:

$$\boldsymbol{v}_{\boldsymbol{max}}^{\boldsymbol{rel}} = \min\left\langle \dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{rend}}, \dot{\boldsymbol{r}}_{\boldsymbol{max}}^{\boldsymbol{cr}} \right\rangle. \tag{28}$$

The next algorithmic step is to determine the instant to switch between interception and rendezvous strategies. This is obtained by finding the instant at which the velocity represented by $\boldsymbol{v}_{max}^{rel}$ can be achieved by the pursuer within the sampling period Δt, namely, the instant at which the velocity $\boldsymbol{v}_{max}^{rel}$ lies within the *Feasible Velocity Region* (*FVR*) as defined in (6).

C. Predictive Obstacle Avoidance

In this work, for the purpose of obstacle avoidance, we have used concepts that are rooted in the missile-guidance/aerospace literature [44]. The motivation behind using such an approach stems from the fact that collision avoidance and collision achievement are, in principle, two aspects of the same problem. Since the proposed obstacle-avoidance method is based on the principles of missile guidance, it allows for an elegant integration with the proposed navigation guidance law used herein (i.e., *APGL*).

Majority of existing dynamic obstacle-avoidance algorithms (e.g., [25]-[30]) attempt to avoid all obstacles which are in the vicinity of the vehicle by evaluating a time-based or distance-based criterion. This may lead to a significant increase in computational complexity in evaluating obstacles which are not on a direct collision course with the vehicle. Thus, the two important decisions in our proposed algorithm are to decide (*i*) whether avoidance is necessary with an obstacle and, if necessary, (*ii*) whether the *APGL* commanded acceleration is sufficient to avoid it.

Obstacle-Avoidance Navigation Law (OANL)

The key objective to collision avoidance is to maintain a predefined safe distance between the pursuer and the obstacle. Let us consider the collision-avoidance problem shown in Figure 5. The pursuer is moving on a 2D plane in the presence of another mobile robot that is designated as an obstacle.

Since for the purpose of the *APGL*, the pursuer is considered to be a point mass, the *OANL* must be based on a similar principle. This is achieved by reducing the pursuer to a point mass and increasing the size of the obstacle by the size of the pursuer. For simplicity, the obstacle is approximated by a circle that envelops the obstacle. Therefore, when increasing the size of the obstacle by the size of the pursuer, one simply increases the radius of the bounding circle of the obstacle by the radius of the pursuer.

The *CC* is defined as a region formed by *APB*. In order to apply the OAN acceleration command, a collision-avoidance vector needs to be defined. This vector may either be represented by ***PA*** or ***PB***. Subsequently, acceleration commands are provided to the pursuer to steer the relative velocity vector between the pursuer and the obstacle towards the collision-avoidance vector.

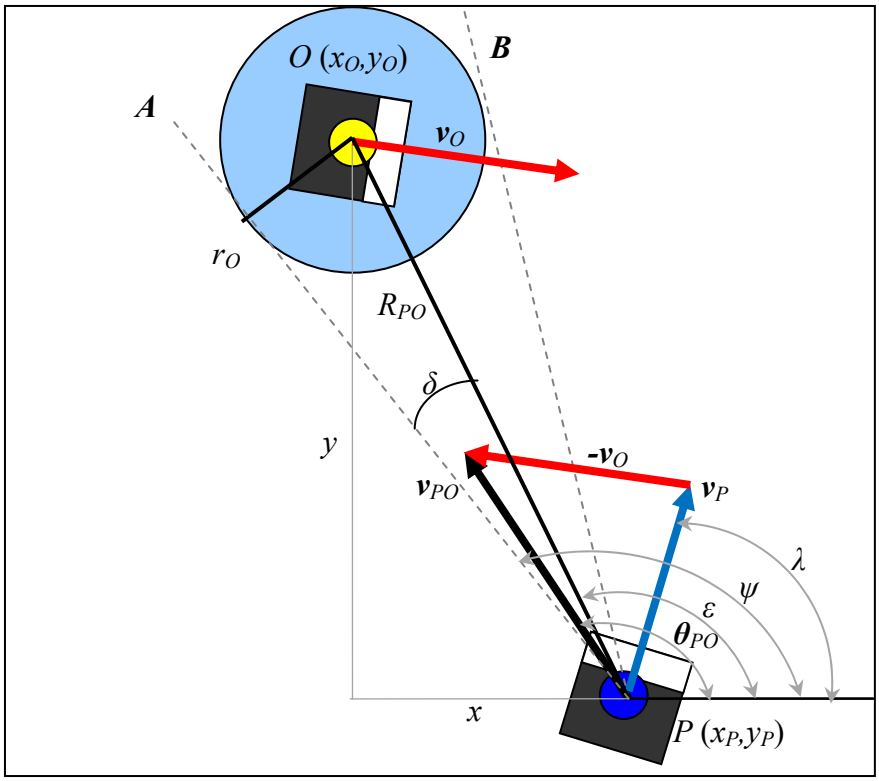


Figure 5. Schematic of the *OANL*.

Based on the geometry in Figure 5, the OAN acceleration command is expressed as:

$$\boldsymbol{a}_c\big|_{OAN} = N\boldsymbol{v}_{\boldsymbol{PO}}\dot{\psi}\ , \tag{29}$$

where $\boldsymbol{a}_c\big|_{OAN}$ is the desired acceleration, $v_{PO}= v_P - v_{PO}$ is the relative velocity, ψ is the direction of the collision-avoidance vector, in this case ***PA***, and N is the navigation constant. From the geometry in Figure 3.2, it can also be concluded that

$$\sin\varepsilon = \frac{y}{R_{PO}}, \quad \cos\varepsilon = \frac{x}{R_{PO}}, \tag{30}$$

$$\sin\delta = \frac{r_O}{R_{PO}}, \tag{31}$$

$$\dot{y} = -v_{PO} \sin\theta_{PO}, \text{ and} \tag{32}$$

$$\dot{\psi} = \dot{\theta}_{PO} + \dot{\varepsilon}. \tag{33}$$

Differentiatin Equations (31) and (32) with respect to time t and substituting into (33) yields,

$$\dot{\psi} = -\left(\frac{v_{PO} \sin\theta_{PO}}{R_{PO} \cos\varepsilon} + \frac{\dot{R}_{PO}}{R_{PO}} \left(\tan\varepsilon + \tan\delta \right) \right). \tag{34}$$

The acceleration for the next time instant using *OANL* is obtained by substituting the above value of $\boldsymbol{a}_c\big|_{OAN}$ into an equation similar to (10):

$$\begin{aligned} \boldsymbol{a}_{OANx} &= -\boldsymbol{a}_c\big|_{OAN} \sin\lambda \\ \boldsymbol{a}_{OANy} &= \boldsymbol{a}_c\big|_{OAN} \cos\lambda \end{aligned}. \tag{35}$$

Prediction of Obstacle Parameters

In this thesis, it is assumed that dynamic obstacles may have different types of motions. Furthermore, as shown in the previous section, the *OANL* depends on an accurate interpretation of the relative velocity between the pursuer and the obstacle. Therefore, in order to predict the movements of highly maneuverable obstacles one needs to accurately track the obstacles. This may be achieved by means of a KF that also considers higher order derivatives in the tracking model for the obstacles. The higher order derivatives in the KF tracking model include obstacle acceleration and jerk. This can be written in a state-space framework as

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \dddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dddot{y} \\ \ddddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w(t), \tag{36}$$

where y, $\dot{y}$, $\ddot{y}$, and $\dddot{y}$ denote the position, velocity, acceleration, and jerk of the target, respectively, and $w(t)$ is the system noise. The measurement vector at the $(k + 1)^{th}$ instant can be expressed in the general form as shown in Equation (37) where $v(t)$ is the measurement noise.

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y}_T \\ \ddot{y}_T \\ \dddot{y}_T \end{bmatrix} + v(t). \tag{37}$$

Since measurements are not taken continuously but, every, $T_{s,}$ seconds, the system model needs to be discretized. The fundamental matrix in discrete form is approximated by Equation (38) which is basically a two term Taylor Series expansion:

$$\varphi_K = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{38}$$

The discrete order process noise matrix is obtained from the continuous process noise matrix according to:

$$Q_k = 2\sigma_p{}^2 \begin{bmatrix} T^7/252 & T^6/72 & T^5/30 & T^4/24 \\ T^6/72 & T^5/20 & T^4/8 & T^3/6 \\ T^5/30 & T^4/8 & T^3/3 & T^2/2 \\ T^4/24 & T^3/6 & T^2/2 & T \end{bmatrix}. \tag{39}$$

In order for the KF to operate, the KF gains K_k need to be calculated. These gains are obtained from a set of recursive Ricatti equations, which are used to yield the KF equations for position, velocity, acceleration and jerk. The process is explained in complete detail in Appendix A.

Collision Prediction

In order for the pursuer robot to make an intelligent decision about when to perform collision avoidance, a four-step procedure that automatically predicts the likelihood of a collision was developed, Figure 6:

Step 1: Check whether the range between the pursuer and the obstacle (R_{PO}) is less than the *safe distance* (R_{Th}), which is a function of a specified *Time Horizon,* T_h, and the velocity. If the answer is yes, proceed to Step 2; otherwise, obstacle avoidance is not required.

Step 2: Check if the direction of the relative velocity φ_{PO} is inside the collision cone. If the answer is yes, proceed to Step 3; otherwise, obstacle avoidance is not required.

Step 3: Check whether the obstacle is not already behind the direction of the relative velocity vector $\varphi_{PO,}$, which would imply that the obstacle has been already avoided. Namely,

check whether $\varphi_{PO} \le -\frac{\pi}{2}\tan^{-1}\left(\frac{y_O - y_P}{x_O - x_P}\right)$ *or* $\varphi_{PO} \ge \frac{\pi}{2}\tan^{-1}\left(\frac{y_O - y_P}{x_O - x_P}\right)$. If the answer is yes, proceed to Step 4; otherwise, obstacle avoidance is not required.

Step 4: Proceed to the obstacle- avoidance algorithm.

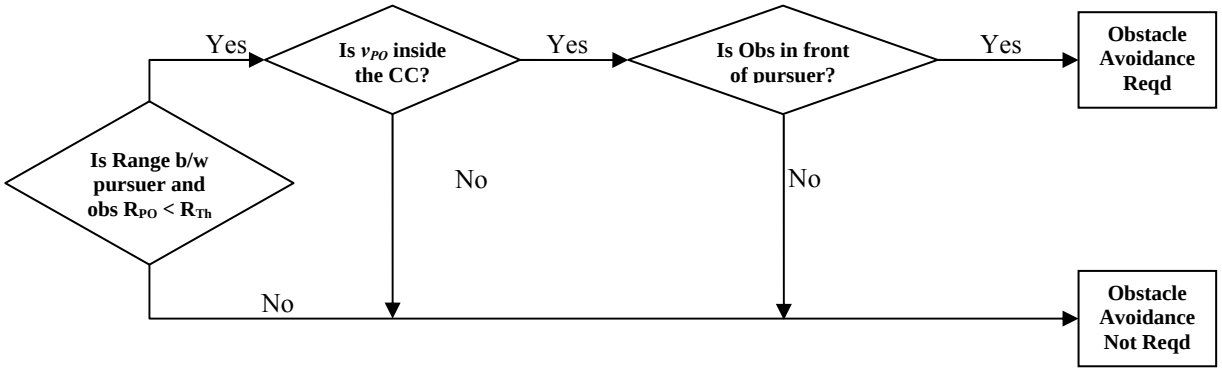


Figure 6: Rules for Collision Prediction.

III. IMPLEMENTATION

In the proposed implementation strategy, for rendezvous with a moving target, the pursuer robot receives its velocity command for the next time instant only from the *APGL* algorithm if obstacle avoidance is not required, Figure 1. The pursuer proceeds in the direction generated by the *APGL*, via $\boldsymbol{a}_{APGL}$, which is only limited by the vehicle's dynamic and kinematic characteristics. Once in the vicinity of obstacles, however, the accelerations generated by the *APGL* algorithm may need to be modified. Collision avoidance takes place based on evaluating the movement of the pursuer and the obstacle. The procedure for generating the acceleration commands for the next discrete time instance is discussed below.

Let us, for example, consider a dynamic environment shown in Figure 7a at a certain instant in time t_i. The pursuer has to rendezvous with a maneuvering target while avoiding a number of dynamic obstacles. It is assumed that the obstacles have similar dynamic characteristics, in terms of speed and maneuverability, as do the pursuer and the target. The procedure for generating the desired acceleration commands for the pursuer is outlined below:

(*i*) Identify and consider only the obstacles within the *safety range*, R_{Th}, associated with the specified time horizon shown in Figure 7a, this would correspond to considering only Obstacle 1 and Obstacle 2 (Figure 7b).

(*ii*) Generate the required acceleration to rendezvous with the target using the *APGL* along the *guidance line,* Figure 7b.

(iii) Determine whether the pursuer can directly move toward the target by using the acceleration command generated in (ii) above or whether obstacle avoidance is required. This is achieved by determining the relative velocity between the pursuer and the obstacle(s) and noting whether it is inside the collision cone between the pursuer and the respective obstacle. In Figure 7b, one can note that avoidance is required for both obstacles.

(*iv*) Determine the accelerations required to avoid the "designated" obstacles individually. In Figure 7b, this corresponds to determining two different accelerations for each obstacle. For example, for Obstacle 1, the first acceleration would steer the relative velocity towards OA_1 that would cause the pursuer to pass in front of the obstacle, whereas the second acceleration steers the relative velocity towards OB_1 that would cause the pursuer to pass behind Obstacle 1. Similarly, two accelerations are determined for avoiding Obstacle 2. The accelerations can be executed by the pursuer by ensuring that the velocities are within the *FVR* parallelogram, Figure 7c.

(*v*) Select an acceleration from (*iv*) such that the pursuer can simultaneously avoid all the obstacles as well as have a minimum deviation form the direction defined by *APGL*. In Figure 7c, for both obstacles, the shortest path to the target would be to pass in front of the obstacles: for Obstacle 1 this would require that the pursuer accelerates to the velocity of v_{P1} whereas for Obstacle 2 it would accelerate to velocity v_{P2}.

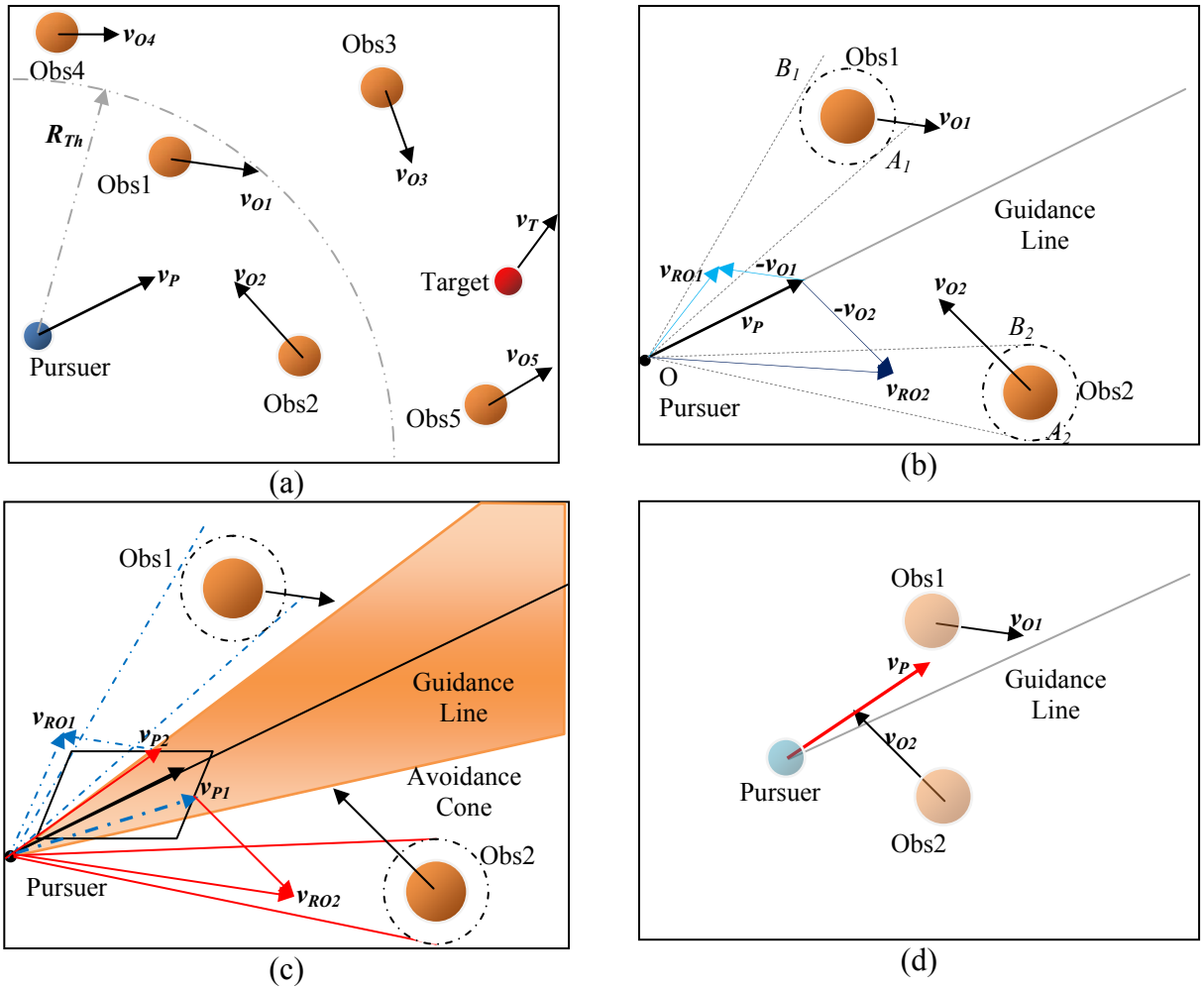


Figure 7: (*a*) Initial Configuration at Time t_i; (*b*) Collision Prediction; (*c*) Avoidance Maneuver; and, (*d*) Final Configuration at Time t_i+dt.

(*vi*) Determine whether accelerating to avoid one obstacle in Step (*v*) above would subsequently lead to a collision with another obstacle. In Figure 7c, for example, in accelerating to velocity v_{P1}, one must ensure that a collision with Obstacle 2 does not result. This check is carried out by determining the relative velocity between v_{P1} and v_{O2} and noting whether it is outside the collision cone. In this example, the relative velocity is within the collision cone, thus, indicating that accelerating to v_{P1} is not suitable for the next time instant since it will cause the pursuer to collide with Obstacle 2. On the other hand, it is similarly determined that accelerating to v_{P2} is a suitable candidate for obstacle avoidance. By accelerating to a velocity that lies outside the combined *avoidance cone*, it is assured that all obstacles are avoided. Furthermore, by choosing a candidate acceleration which changes the velocity of the pursuer to a velocity outside the *avoidance*

cone closest to the rendezvous velocity obtained from *APGL* provides the desired optimal velocity for the next time instant. In this example, it is v_{P2}, which would cause the robot to pass in front of Obstacle 2, but, behind Obstacle 1, Figure 7d.

A primary advantage of using the above procedure is the consideration of only the obstacles that would potentially collide with the pursuer while obtaining a time-optimal trajectory. Thus, the acceleration of the pursuer $\boldsymbol{a_P}$ for the next time instant (t_i+dt) is given by:

$$\boldsymbol{a_P}(t_i + dt) = \begin{cases} \boldsymbol{a_{APGL}} & \textit{if obstacle avoidance is not required} \\ \boldsymbol{a_{OANL}} & \textit{if obstacle avoidance is required.} \end{cases} . \tag{40}$$

IV. SIMULATIONS

A number of simulations were carried out using the proposed *APGL*-based algorithm, Table 1: the maximum velocity and lateral acceleration of the pursuer robot was limited to 200 mm/s and 2000 mm/s^2, respectively, in all the examples, and the criterion for successful rendezvous was set to <10 mm relative distance in both *X* and *Y* directions and a relative velocity of <10 mm/s. Furthermore, in order to examine the effect of noise on the performance of the proposed algorithm, simulated noise (up to 5%) was added to the target's "measured" position: 1% noise is equivalent to about 10 mm in robot travel.

Simulation Results

Figure 8a shows the results of a simulation carried out with static obstacles. Figures 8b, 8c, and 8d, in turn, show simulations in which both the obstacles and the target are moving in straight-line, circular, or sinusoidal paths.

The simulations verified that (in all scenarios, including cases where the obstacles were moving at considerable speeds) the proposed algorithm can effectively and time-optimally avoid collisions by deciding on the best direction to navigate in order to rendezvous with the target. In dealing with static obstacles, Figure 8a, the pursuer recognized that only a minor deviation is required to navigate around both obstacles and achieve an optimal rendezvous with the target. In the scenarios depicted in Figures 8b, 8c, and 8d, the algorithm correctly decides whether to accelerate in front of the obstacle or to pass it from the rear based on an accurate prediction of the future behavior of the obstacles (position, velocity, and acceleration) as well as of the target.

Table 1: Summary of Simulation Data.

S.No	*Obstacle 1*		*Obstacle 2*		*Target*		*Rendezvous Time* (s)
	Type	*Max. Vel* (mm/s)	*Type*	*Max. Vel* (mm/s)	*Type*	*Max. Vel* (mm/s)	
1	*Static*	0	*Static*	0	*Sinusoidal*	120	8.2
2	*Straight*	150	*Straight*	150	*Straight*	130	7.9
3	*Circular*	150	*Circular*	100	*Sinusoidal*	110	8.8
4	*Sinusoidal*	180	*Sinusoidal*	170	*Sinusoidal*	120	8.5

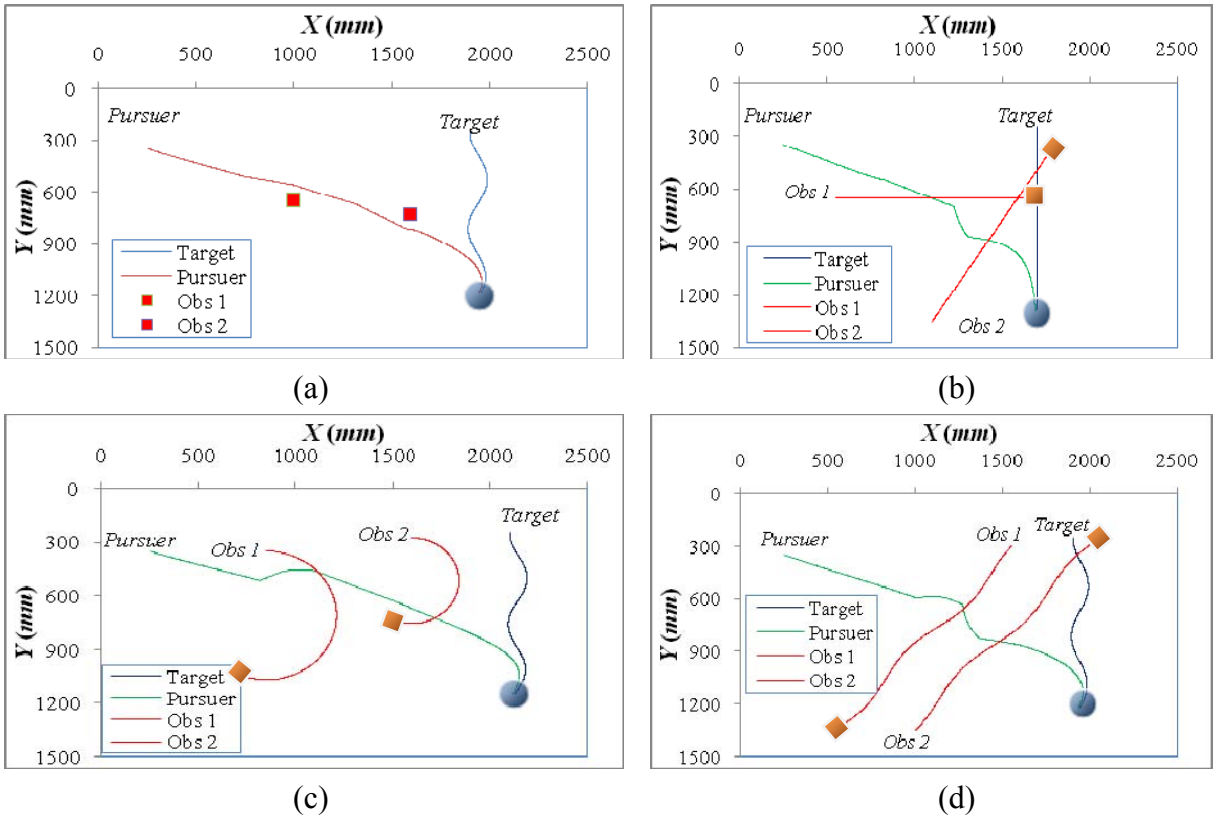


Figure 8: Simulations with (*a*) Static Obstacles, (*b*) Obstacles and Target Moving in Straight Lines, (*c*) Obstacles Moving on Circular Paths, and (*d*) Obstacles Moving Sinusoidally.

V. EXPERIMENTS

The physical layout of the experimental set-up is depicted in Figure 9 and the hardware specifications are given in Table 2. The software for the experiments, running on a Pentium IV 1.6 GHz processor PC, consisted of three modules: image acquisition and processing, trajectory planning, and communication modules, respectively. An analog CCD camera captured the image

of the workspace and transferred it to the frame-grabber in the PC. The vision algorithm, then, extracted the positional information of all the objects in the workspace. This information was sent to the trajectory planner, where an acceleration command is calculated for the robot/pursuer. The communication module broadcasted this data to the mobile robots via a *Bluetooth* module connected to the PC. The details of the vision system, communication system and mobile robots are included in Appendix B.

Table 2: Experimental Hardware.

Component	***Characteristics***
Mobile Robots	Miabot Pro Wireless (Bluetooth) Controlled
PC	Host computer, frame grabber and RF module
CCD Camera	Resolution: 640×480 pixels Lens focal length: 6 mm Distance from floor: 3000 mm
Floor Workspace	2740×1500 mm Surface material: Felt

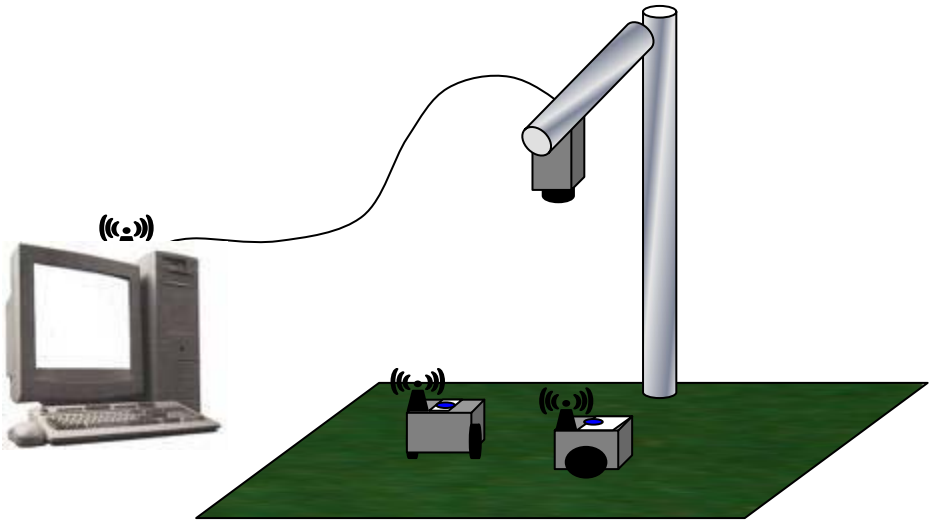


Figure 9: Physical Layout of the Set-up.

Experimental Results

Experiments were carried out with the aim to intercept a moving target without trying to rendezvous with it (for equipment safety reasons). In the first experiment presented here, the target is moving on a straight line and the obstacles are static (Figure 10). In the remaining experiments, one obstacle remains static and the other obstacle and the target are mobile, Table 3

and Figures 11 to 13. Each experiment was repeated three times under identical conditions. Experimental results show that the real-time paths followed by the pursuer in the experiments are similar to its paths observed in simulation.

Table 3. Summary of Experimental Data

Trial	***Pursuer Vel.*** (mm/s)	***Tgt. Vel*** (mm/s)	***Obs. 1 Vel.*** (mm/s)	***Obs. 2 Vel.*** (mm/s)	***Mean Interception Time*** (s)
Static	200	100	0	0	6.73
Linear	200	125	100	0	8.13
Circular	200	100	130	0	11.73
Sinusoidal	200	80	80	0	8.10

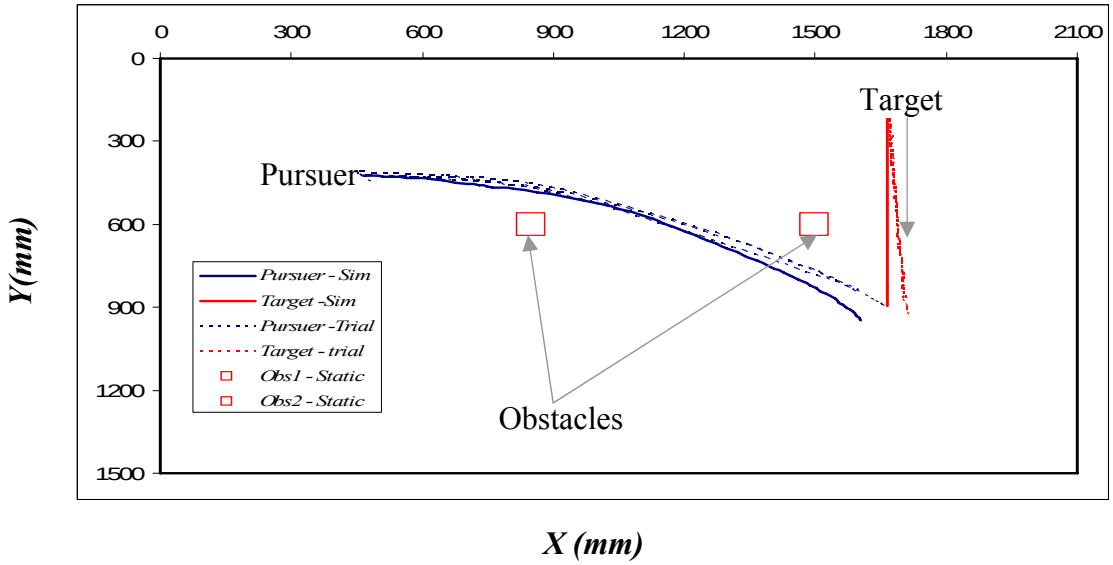
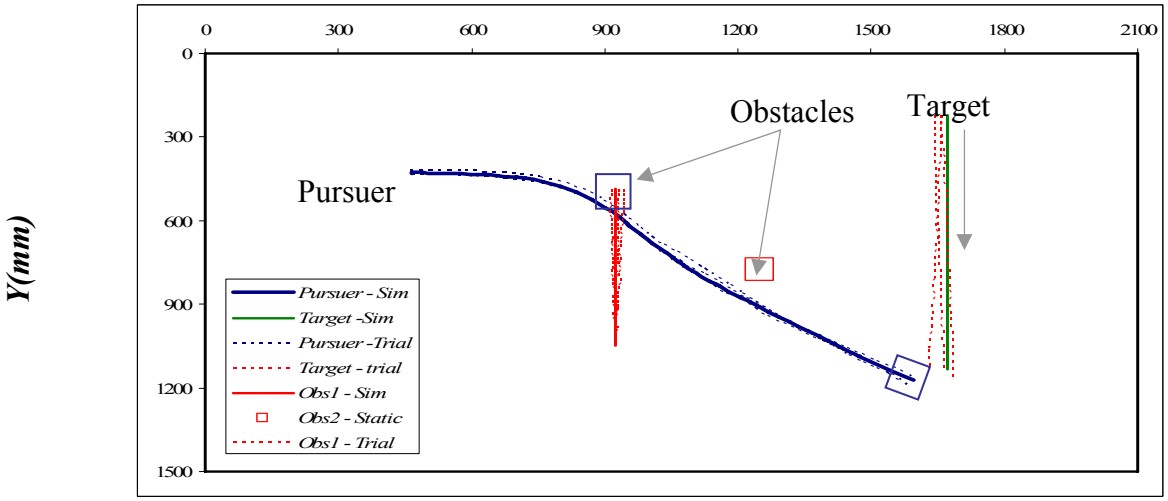
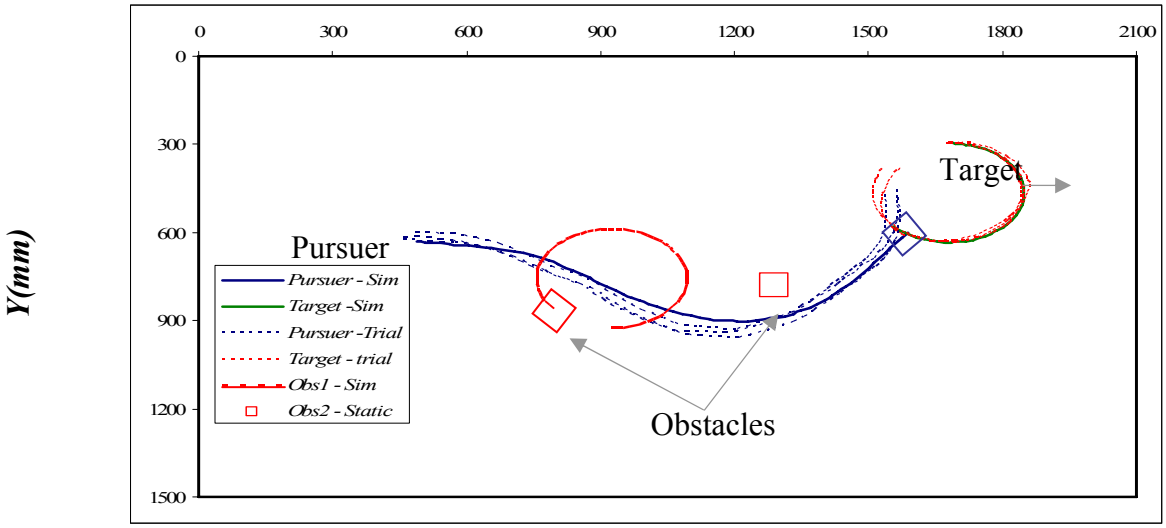


Figure 10: Experiments with A Static Obstacle.



X (mm)

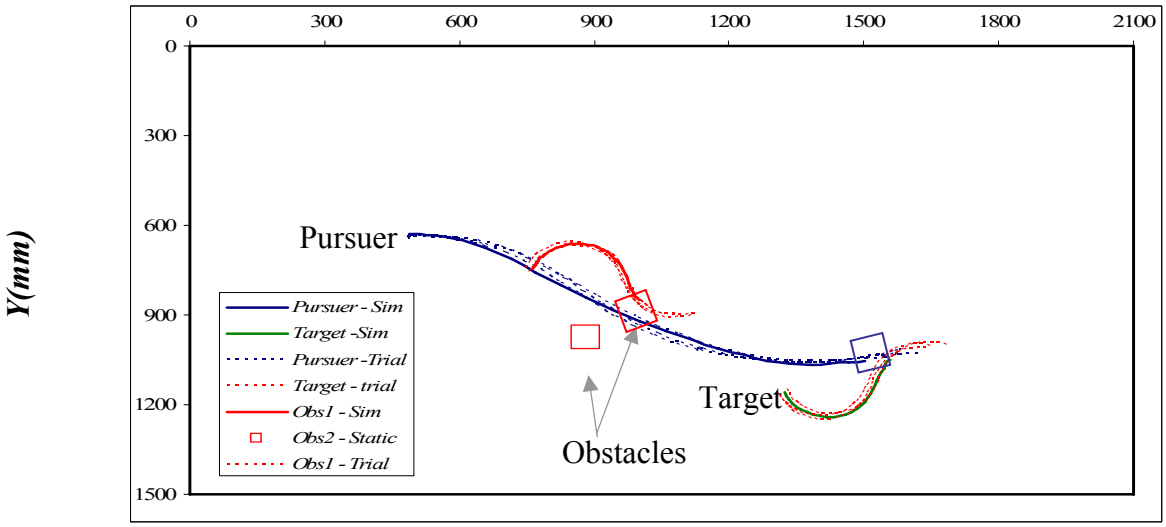
Figure 11: Experiments with Linear-Motion Obstacle and Target.



Y(mm)

X (mm)

Figure 12: Experiments with Circular Motion Obstacle and Target.



Y(mm)

X (mm)

Figure 13: Experiments with Sinusoidal Motion Obstacle and Target.

VI. CONCLUSIONS

A novel rendezvous-guidance method is proposed for autonomous robotic interception of highly maneuvering targets in a dynamic environment with static and/or dynamic obstacles. The future

maneuver of the target is predicted using a five-state Extended Kalman Filter (EKF). The proposed algorithm, then, uses the novel Advanced Predictive Guidance Law (*APGL*) to obtain the required acceleration commands for rendezvous with the target. In the presence of obstacles, the algorithm uses the novel Obstacle-Avoidance Navigation Law (*OANL*), which first predicts the likelihood of a collision and, then defines a collision cone for each potentially colliding obstacle (within close proximity of) the pursuer. Based on this information the algorithm directs the relative velocity between the pursuer and the obstacle outside the *collision cone*. By employing a velocity and heading outside the *collision cone*, a collision-free trajectory to the target is ensured. EKF is also used to track the obstacles states.

Furthermore, in our algorithm, instead of using some form of a heuristic search strategy, proposed in most obstacle-avoidance techniques, the search for a feasible velocity for the next sampling interval is reduced to velocities that are as close to the maximum closing velocity component obtained from the *APGL* method as possible.

Simulations and experiments have verified the system to be efficient and robust in regards to interception of moving targets with various different interception parameters and situations.

REFERENCES

[1] F. Kunwar, F. Wong, R. Ben Mrad, and B. Benhabib, "Rendezvous Guidance for The Autonomous Interception of Moving Objects in Cluttered Environments," *IEEE Conference on Robotics and Automation*, pp. 3787-3792, Barcelona, Spain, April 2005.

[2] F. Kunwar, F. Wong, R. Ben Mrad, and B. Benhabib, "Time-Optimal Rendezvous with Moving Objects in Dynamic Cluttered Environments Using A Guidance Based Technique," *IEEE International Conference on Intelligent Robots and Systems,* pp. 283-288, Edmonton, Canada, August 2005.

[3] Zarchan P., **Tactical and Strategic Missile Guidance**, Fourth Edition, American Institute of Aeronautics and Astronautics, Virginia, 2003.

[4] H. L. Pastrick, S. M. Seltzer, and M. E. Warren, "Guidance Laws for Short-Range Tactical Missiles," *Journal of Guidance, Control and Dynamics*, Vol. 4, No. 2, pp. 98-108, 1981.

[5] G. M. Anderson, "Comparison of Optimal Control and Differential Game Intercept Missile Guidance Law" *AIAA Journal of Guidance and Control*, Vol. 4, No. 2, pp.109-115, March 1981.

[6] D. Ghose, "True Proportional Navigation with Manoeuvring Target," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 1, No. 30, pp. 229-237, January 1994.

[7] T. J. Speyer, K. Kim, and M. Tahk, "Passive Homing Missile Guidance Law Based on New Target Maneuver Models," *Journal of Guidance*, Vol. 1, No. 13, pp. 803-812, September 1990.

[8] C. D. Yang, and C. C. Yang, "A Unified Approach to Proportional Navigation," *IEEE, Transactions on Aerospace and Electronic Systems,* Vol. 33, No. 2, pp. 557-567, 1997.

[9] P. J. Yuan, and S. C. Hsu, "Rendezvous Guidance with Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 2, pp. 409-411, 1993.

[10] M. Guelman, "Guidance for Asteroid Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, pp. 1080-1083, 1990.

[11] D. L. Jensen, "Kinematics of Rendezvous Manoeuvres," *Journal of Guidance*, Vol. 7, No. 3, pp. 307-314, 1984

[12] M. Mehrandezh, M. N. Sela, R. G. Fenton, and B. Benhabib, "Robotic Interception of Moving Objects Using an Augmented Ideal Proportional Navigation Guidance Technique," *IEEE, Transactions on Systems, Man and Cybernetics*, Vol. 30, No. 3, pp. 238-250, 2000.

[13] J. M. Borg, M. Mehrandezh, R. G. Fenton, and B. Benhabib, "Navigation-Guidance-Based Robotic Interception of Moving Objects in Industrial Settings," *Journal of Intelligent and Robotic Systems,* Vol. 33, No. 1, pp. 1-23, 2002.

[14] F. Agah, M. Mehrandezh, R. G. Fenton, and B. Benhabib, "On-line Robotic Interception Planning Using Rendezvous-Guidance Technique," *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 40, No. 1, pp. 23-44, May 2004.

[15] U. Ghumman, F. Kunwar, and B. Benhabib, "Guidance-Based On-Line Motion Planning for Autonomous Highway Overtaking," *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1, No. 2, pp. 589-571, June 2008.

[16] Latombe, J.C., **Robot Motion Planning**, Boston: Kluwer Academic Publishers, 1991.

[17] J. Baraquand, B. Langlois, and J. C. Latombe, "Numerical Potential Field Techniques for Robot Path Planner," *IEEE*, *Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 2, pp. 224-241, March-April 1992.

[18] J. Borenstein, and Y. Koren "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278-288, June 1991'

[19] J.P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray, "A Motion Planner for Nonholonomic Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 5, pp. 577-593, October 1994.

[20] Laumond, J.P., **Robot Motion Planning and Control**, Springer-Verlag Telos, 1998.

[21] R. Simmons, "The Curvature-Velocity Method for Local Obstacle Avoidance", *IEEE International Conference on Robotics and Automation,* pp. 2275-2282, Minneapolis, MN, April 1996.

[22] D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," *IEEE Robotics and Automation Magazine*, Vol. 4. No.1, pp. 23-33, March 1997.

[23] W. Feiten, R. Bauer, and G. Lawitzky, "Robust Obstacle Avoidance in Unknown and Cramped Environments," *IEEE International Conference on Robotics and Automation,* pp. 2412-241, May 1994.

[24] F. Zhang, A. O'Conner, D. Luebke, and P.S. Krishnaprasad, "The Experimental Study of Curvature-Based Control Laws for Obstacle Avoidance," *IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3849-3854, New Orleans, LA, April, 2004.

[25] N.Y. Ko and R.G. Simmons. "The Lane-Curvature Method for Local Obstacle Avoidance," *IEEE Conference on Intelligent Robots and Systems*, pp. 1615- 1621 Victoria, Canada, October 1998.

[26] P. Ogren and N.E. Leonard, **"**A Tractable Convergent Dynamic Window Approach to Obstacle Avoidance," *IEEE International Conference on Intelligent Robots and Systems,* pp. 595-600, Lausanne, Switzerland, September 2002.

[27] H. Hu and M. Brady, "A Bayesian Approach to Real-time Obstacle Avoidance for Mobile Robots," *Autonomous Robots*, Vol. 1, pp. 69-92, January 1994.

[28] D. Fox, W. Burgard., S. Thrun and A. Cremers, "A Hybrid Collision Avoidance Method for Mobile Robots", *IEEE International Conference on Robotics and Automation*, pp. 1238-1243, 1998.

[29] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments using Velocity Obstacles," *International Journal on Robotics Research*, Vol. 17, No. 7, pp. 711-727, 1998.

[30] F. Large, S. Sekhavat, Z. Shiller,and C. Laugier, "Towards Real-Time Global Motion Planning in a Dynamic Environment Using the NLVO Concept," *IEEE International Conference on Intelligent Robots and Systems*, pp. 607-612, Lausanne, Switzerland, 2002.

[31] Z. Li, J. Canny, J. and G. Heinzinger, "Robot Motion Planning with Non-Holonomic Constraints." *5th International Symposium of Robotics Research,* pp. 343-350. Tokyo, Japan, 1989.

[32] P. Rouchen, M. Fliess, J. Levine, and P. Martin, "Flatness and Motion Planning: the Car with *n* Trailers," European Control Conference, pp. 1518-1522, 1992.

[33] R. M. Murray and S. S. Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids,'' *IEEE, Transactions on Automatic Control,* Vol. 38, pp. 700-716, 1993.

[34] *S.* Monaco and D. Normand-Cyrot, "An Introduction to Motion Planning under Multirate Digital Control," *IEEE Conference on Decision and Control,* Tucson, Arizona, pp. 1780-1785, December 1992.

[35] D. Elbury, R. M. Murray, and S. S. Sashy, "Trajectory Generation for the *n*-Trailer Problem Using Goursat Normal Form," *IEEE, Transactions on Automatic Control,* Vol. 40, pp. 802-819, 1995.

[36] C. Fernandes, L. Gurvits, and 2. *Li,* "Near-optimal Nonholonomic Motion Planning for A System of Coupled Rigid Bodies," *IEEE, Transactions on Automatic Control,* Vol. 39, pp. 450463, 1994.

[37] D. J. Balkcom, and M. T. Mason, "Extrema1 Trajectories for Bounded Velocity Differential Drive Robots," *IEEE International Conference on Robotics & Automation*, pp. 1747-1752, California, April 2000.

[38] D. J. Balkcom, and M. T. Mason, "Extrema1 Trajectories for Bounded Velocity Mobile Robots," *IEEE International Conference on Robotics & Automation*, pp. 1747-1752, Washington, DC, May 2002.

[39] S. Sundar and Z. Shiller, "Optimal Obstacle Avoidance Based on the Hamilton-Jacobi-Bellman Equation," *IEEE, Transactions on Robotics and Automation.* Vol. 13, pp. 305-310, 1997.

[40] A. E. Bryson and Y.-C. Ho, **Applied Optimal Control***,* 2nd Edition, Hemisphere Publishing Corporation, New York, 1975.

[41] Z. Qu and J. R. Cloutier, "A New Suboptimal Control Design for Cascaded Nonlinear Systems," *Optimal Control: Applications and Methods,* vol. 23, pp. 303-328, 2002.

[42] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A Motion Planner for Nonholonomic Mobile Robots," *IEEE Transactions on Robotics and Automation,* Vol. 10, pp. 577-593, 1994.

[43] J. A. Reeds and R. A. Shepp, "Optimal Paths for a Car that Goes Both Forward and Backwards," *Pacific J. Mathematics,* Vol. 145, pp. 367-393, 1990.

[44] P. Vadekkapat, and D. Goswami, "Biped Locomotion: Stability Analysis and Control," *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1, No.1, pp. 589-571, March 2008.

[45] A. Chakravarthy and D. Ghose, "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach," *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans*, Vol. 28, No. 5, September 1998.

[46] D. Bourgin, "Color Space FAQ," http://www.neuro.sfc.keio.ac.jp/~aly /polygon/info/ color-space-faq.html, August 2004.

[47] C.B. Bose and J. Amir, "Design of Fiducials For Accurate Registration Using Machine Vision," *IEEE*, *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 12, pp. 1196-1200, December 1990.

APPENDIX A: EXTENDED KALMAN FILTER

The Kalman filter is a two-step probabilistic estimation process that is very popular in the robotics world as a tool to predict the next position of the robot in a linear system. Kalman filters are based on linear algebra and the hidden Markov model. The underlying dynamical system is modeled as a Markov chain built on linear operators perturbed by Gaussian noise. The state of the system is represented as a vector of real numbers. At each discrete time increment, a linear operator is applied to the state to generate the new state, with some noise mixed in, and

optionally some information from the controls on the system if they are known. Then, another linear operator mixed with more noise generates the visible outputs from the hidden state. The Kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state.

The *Extended Kalman Filter* (EKF) is similar to the KF but it can be used in non-linear systems because it linearizes the transformations via the Taylor Expansions. In the EKF the state transition and observation models need not be linear functions of the state but may instead be (differentiable) functions. A generalized EKF is shown in Figure A1.

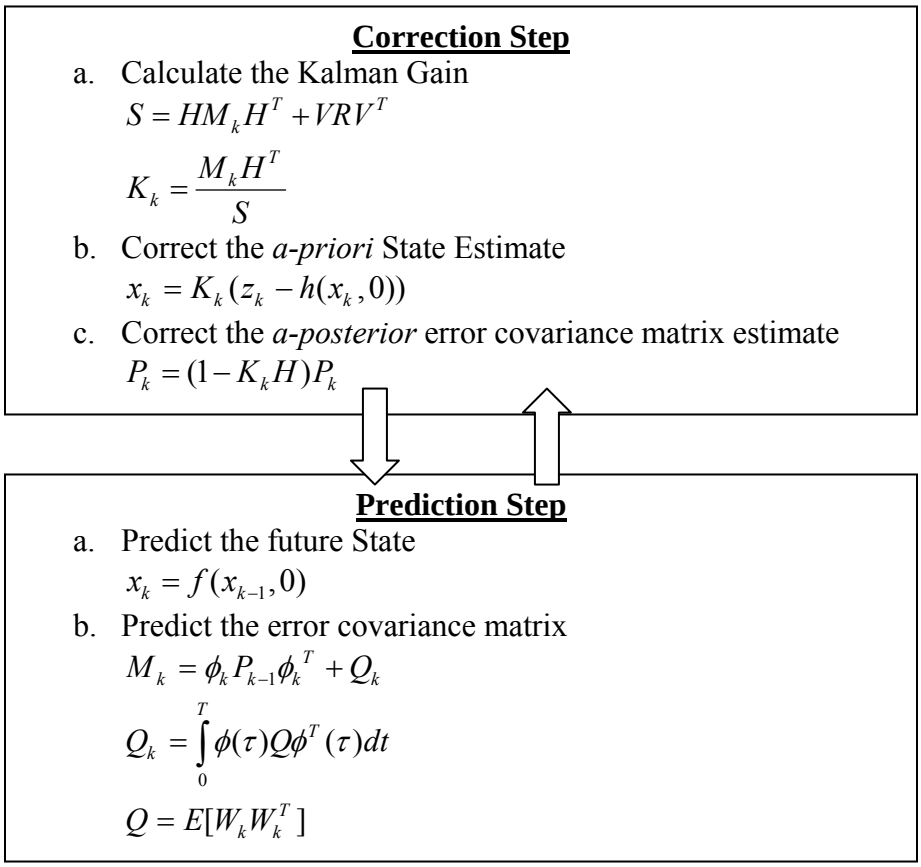


Figure A1: Extended Kalman Filter.

Notation

$\boldsymbol{x}$ state estimate
z measurement data
$\boldsymbol{\phi}$ Jacobian of the system model with respect to state
$\boldsymbol{W}$ Jacobian of the system model with respect to process noise
$\boldsymbol{V}$ Jacobian of measurement model with respect to measurement noise
$\boldsymbol{H}$ Jacobian of the measurement model
$\boldsymbol{Q}$ process noise covariance
$\boldsymbol{R}$ measurement noise covariance
$\boldsymbol{K}$ Kalman Gain
$\boldsymbol{P}$ estimated error covariance
σ_p prediction noise
σ_m measurement noise

APPENDIX B: DETAILS OF THE EXPERIMENTAL SET-UP

B1. Vision System

The robot, the obstacles, and the target were color-coded for identification. The raw image containing three channels of data, indicating the intensities of the Red (*R*), Green (*G*) and Blue (*B*) colors, in each pixel were transformed into the *YCbCr* (luminance, chrominance-blue, and chrominance-red) color space. The transformation is performed by [47]:

$$Y = 0.299R + 0.587G + 0.114B \tag{B1}$$

$$Cb = (B - Y)/1.772 \tag{B2}$$

,

$$Cr = (R - Y)/1.402 \tag{B3}$$

where *Y* has a range of [0, 255] and *Cb* and *Cr* both have a range of [−127.5, 127.5].

When an image is examined, the weighted Euclidean distances, in the *YCbCr* color space, between each pixel in the image and the predefined colour set are calculated:

$$D = \sqrt{0.15(Y_p - Y_c)^2 + 0.425(Cb_p - Cb_c)^2 + 0.425(Cr_p - Cr_c)^2}\ , \tag{B4}$$

where *D* is the weighted Euclidean distance, Y_p, Cb_p, and Cr_p, are the measured *YCbCr* values of the pixel, and Y_c, Cb_c, and Cr_c are the values of the predefined color set. During the experiments, it was noted that the pixels on the identification marker did not vary more than 18.0 in weighted Euclidean distance from the defined *YCbCr* value. This value was, therefore, set as the threshold

distance: if a pixel is within this threshold distance of a certain color in the predefined color set, then, it is considered to be that color. After the image has gone through the thresholding operation, the positions of the mobile robot, the obstacles, and the target are determined.

A search is performed to find the markers on all the objects. In order to achieve the smallest sampling rate, the dimension of the smallest marker is used, denoted here as l pixels. Starting from the pixel location (0, 0), every $0.5l$ pixels are sampled along the X and Y directions. If the sampled pixel has the color of the predefined set, a search frame is placed over that pixel. The size of the search frame is twice the diameter of the marker. If the number of pixels of a certain color in the search frame exceeds a pre-determined threshold, then, a marker of that color is considered to be located in that search frame. The centroid of that color blob is, then, calculated to sub-pixel accuracy using the Centroid Method [46], Figure B1.

With all the markers located, object identification can be performed. The vision program first searches for blue markers. Once a blue marker is found, the algorithm looks for a white marker within a distance of the radius of a robot. If a corresponding white marker is located, then, a robot has been successfully identified. Bearing of the object is indicated by an imaginary line drawn from the centre of the blue circle to the centroid of the white pattern. The algorithm takes approximately 150 ms to execute (i.e., a frame-rate of 6.5 fps).

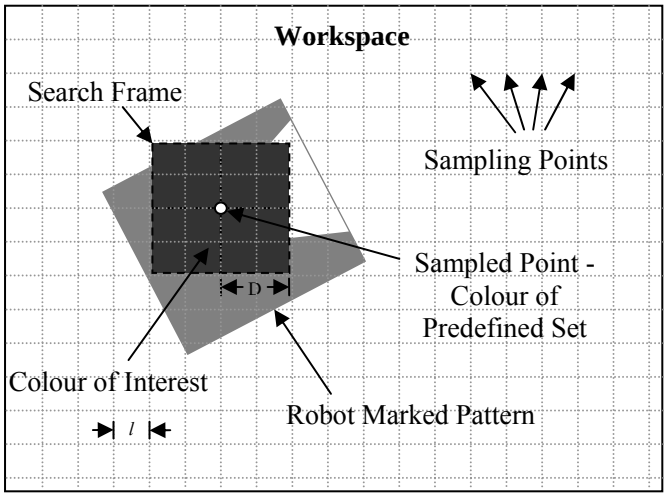


Figure B1: Color Marker Search.

B2. Communication System

The Bluetooth card enables the robot to communicate with the host PC, converting the Bluetooth link to logic-level serial signals. The MIABOT-BT Bluetooth boards are supplied with fixed communication settings 19200 baud (8 bits, 1 stop bit, no parity). A PC Bluetooth dongle is supplied that plugs into the USB port on the PC. This can support wireless links with up to 7 robots at once.

B3. Mobile Robots

Three **Miabot PRO BT v2** differential-drive mobile robots were used in the implementation of the proposed methodology: a pursuer, a moving obstacle, and a target. The robot motors are driven by 6×1.2 V (AA) cells through a low-resistance driver I.C. with a slow-acting current limit at about 5A. Maximum speed of an unloaded motor is in the region of 6000 to 8000 rpm. The motor shafts drive the wheels through an 8:1 gearing. The motors incorporate quadrature encoders giving 512 position-pulses per rotation. The wheels are 52mm in diameter; one encoder pulse corresponds to just under 0.04 mm of movement. Each robot contains two 3 AA-cell battery packs (nominal 1.2V per cell, 1300 mAh).

ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).