LOCALIZATION AND CONSTRAINED NON-LINEAR OPTIMAL CONTROL IN AUTONOMOUS SYSTEMS

by

Anuradha Viswanathan

Master of Science in Telecommunications

University of Pittsburgh

2010

Submitted to the Graduate Faculty of

School of Information Science in partial fulfillment

of the requirements for the degree of

Master of Science in Telecommunications

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

School of Information Science

This thesis was presented

by

Anuradha Viswanathan

It was defended on

19 January, 2010

and approved by

Prashant Krishnamurthy, Associate Professor, School of Information Sciences

John Dolan, Senior Systems Scientist, Robotics Institute, CMU

David Tipper, Associate Professor, School of Information Sciences

Thesis Director: David Tipper, Associate Professor, School of Information Sciences

LOCALIZATION AND CONSTRAINED NON-LINEAR OPTIMAL CONTROL IN AUTONOMOUS SYSTEMS

Anuradha Viswanathan, M.S.

University of Pittsburgh, 2010

Path planning for an autonomous vehicle in a dynamic environment is a challenging problem particularly if the vehicle has to utilize its complete maneuvering abilities, perceive its environment with a high degree of accuracy and react to unsafe conditions. Trajectory planning in an environment with stationary and moving obstacles has been the topic of considerable interest in robotics communities and much of the work focuses on holonomic and non-holonomic kinematics. Optimal path planning has been approached using numerical optimization techniques planning the navigation of ground and aerial navigation producing realistic results in spite of computational complexity. Most of the previous work discussed uses static obstacles and autonomous vehicles moving in closed indoor environments involving prior knowledge of its environment using map based localization and navigation. The work that has focused on dynamic environments with moving obstacles having assumptions of completely known velocities don't account for uncertainty during obstacle motion prediction. Estimation based approaches use grid-based environment representation of the state space, discretized velocities and linear motion models. This simulation aims at finding an optimal trajectory by obtaining the optimal longitudinal and lateral maneuvers using the vehicle's sensing and predictive capabilities for path planning in continuous 2-D space. The focus is on specific scenarios using spatial and temporal constraints while navigating and it involves timed maneuvering in between periods of

straight line motion as for a typical unmanned ground vehicle. It also combines tracking obstacles independently and relative localization with targets to achieve its objective. The parametric space of longitudinal and lateral velocities is generated for the host vehicle aiming to reach a goal state configuration within a pre-specified time threshold. This considers independently the cases for completely known trajectories of obstacles and motion under uncertainty. The results of constrained non-linear optimization allow the vehicle to trace its trajectory given its known initial and destination configuration along with known velocity profiles, noise models and range-bearing measurements to the targets in its vicinity. Simulation results show that the proposed scenario-specific approaches produce reasonable maneuvers within the admissible velocity ranges.

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

ACKNOWLEDGEMENTS

First and foremost, I want to dedicate this to my parents and sister without whose emotional support this thesis would have been impossible for me. My sister has gone through a lot of pain and sacrifice to help me along with my life and she is my reason and motivation for any future endeavors. I would like to thank Dr. John for his willingness to work with me and his immensely valuable support and enthusiasm without which I would not have been able to do this thesis.

I would like to thank Dr. Prashant for his encouragement and being the greatest source of inspiration in doing this thesis and in all academic pursuits and I would like to thank Dr. Thompson for being the biggest motivating factor for me choosing to pursue my Master's degree in this university. I would like to thank Dr. Tipper for playing the role of a very supportive academic advisor and Dr. Kabara for the very enjoyable experiences in class and his enthusiasm and confidence boosting while sending me to my first conference in U.S.

I would like to thank some of my PH.D peers as Pat, Korporn, Maria and Yuttasart who have helped me with different things at different stages and special acknowledgements to a couple of my batchmates for being amazing project/lab mates and for being of great help in initially settling down in Pittsburgh. Last but not the least, I would like to thank the staff for their prompt and valuable help with various official issues which allowed me to have a smooth experience at Pitt.

1.0 INTRODUCTION

Various collision detection and motion planning algorithms are designed to report a geometric contact between moving objects such as polygons, splines and algebraic surfaces. It is widely employed in autonomous vehicle motion planning and control and is very popular in the video gaming industry and virtual reality applications, CAD/CAM and for computer graphics and animation. Planning algorithms have been widely successful in several academic disciplines including robotics, multi-agent planning and aerospace applications. A survey paper presented gives an interesting array of approaches to collision detection and the geometric models investigated for the same purposes.

Collision avoidance technologies have been a popular area of research since these technologies have been applied to several scenarios involving motor vehicles, robots, airplanes and submarines. A lot of focus has been on cooperative vehicular collision avoidance system which has been used in conjunction with adaptive cruise control systems. Originally vehicular collision avoidance systems were used for automated highway systems to increase capacity of freeways. Optical collision avoidance systems have also been very popular since they use image processing algorithms that help to identify objects. The dynamics of these objects are estimated using tracking algorithms so that a warning can be issued if collision is imminent. The other technologies involved for the same purpose include the use of infrared vehicle-mounted cameras to track movements of objects surrounding the subject vehicle.

The basic motion planning problem constitutes finding an a priori set of configurations that leads an autonomous system to a predefined objective. Configuration is defined as the set of parameters that represents the position, orientation and motion primitives of the system.

The motion plan often depends on the models of the system characteristics as well as the environment in which it operates. However, the traditional path planning approaches consider the autonomous system navigating amidst stationary obstacles and are geometric representations of the system and obstacles were sufficient for path formulation. In the case of time-varying dynamic environments in realistic applications, planning must take into consideration spatio-temporal aspects which require time-parameterization of the sequence of configurations. This introduces the need for trajectory parameterization and in addition to considering temporal aspects, the constraints based on kinematics and dynamics must also be considered for autonomous vehicle motion. The models used for motion planning must also account for the uncertainties in motion and sensing characteristics in the autonomous vehicle and must be incorporated at the planning level which adds to the complexity of the problem. The main objective of this implementation is to provide a prototype of autonomous vehicle navigation considering several constraints to trajectory planning and optimal control such as time to destination, obstacle avoidance and in addition to these, includes specific real-time constraints applicable to the context of advanced driver assistance systems. The architecture of the application relies on laser rangefinders for relative localization with obstacles, dynamic characteristics of obstacles such as position, velocity and future trajectories and scenario-specific trajectory optimization by the host vehicle.

1.1 OPTIMAL CONTROL PROBLEM

Optimal control is an essential in a large number of applications and has played an important role in developing the design criteria for modern systems. The class of autonomous systems that are considered include rockets, airplanes, ground vehicles and submarines each of which have different application requirements, economic and environmental constraints to carry out their mission. The measure of performance of the system is defined as an objective function satisfying a set of constraints and the optimal control signals are determined to satisfy the constraints as well as maximize or minimize the objective. The formulation of the optimal control problem consists of:

1. A mathematical model of the system
2. The set of constraints
3. Performance criteria defined by the objective function

1.1.1 Mathematical model

The first part of the problem consists of defining the simplest mathematical model that can accurately predict the response of the system's state given the control inputs. If the state variables of the system at time t are given by the state vector $\begin{pmatrix} x1(t) \\ \vdots \\ xn(t) \end{pmatrix}$ and the control vector as $\begin{pmatrix} u1(t) \\ \vdots \\ um(t) \end{pmatrix}$, the state equations for the class of non-linear time-invariant systems can be written as,

$$\dot{x}(t) = a(x(t), u(t))$$

where a describes the n first-order differential equations given by,

$$\dot{x}1(t) = a1\big(x1(t), x2(t),,,, xn(t), u1(t), u2(t),,,,, um(t)\big)$$

$$\dot{x}2(t) = a2\big(x1(t), x2(t),,,, xn(t), u1(t), u2(t),,,,, um(t)\big)$$

$$\vdots$$

$$\dot{x}n(t) = an\big(x1(t), x2(t),,,, xn(t), u1(t), u2(t),,,,, um(t)\big)$$

1.1.2 Constraints specification

Once the mathematical model is defined, it is necessary to specify the constraints on the state and the controls. These could include the initial and final state, time to destination, upper and lower bounds on the control inputs such as lateral and longitudinal velocity and acceleration. A state trajectory which satisfies the state variable constraints throughout the time interval is known as the admissible trajectory which is important since it reduces the search space of allowable controls and states used by the system. This can be formally stated as finding an admissible control vector which leads to an admissible trajectory and minimizing the objective function.

1.1.3 Performance criteria

Performance measures used to design systems depend on various factors and on what the designer prioritizes for the application to perform as desired. Some typical control problems that provide a necessity for selecting performance measures include

1. Minimum-time problems such as transferring a system from an initial state to a specified target state in minimum time
2. Terminal control problems which require to minimize the deviation of the final state of the system from its desired value
3. Minimum-control-effort problems which transfer a system from an initial to the final configuration with a minimum expenditure of control that depends on the particular application such minimizing fuel consumption, energy dissipation on a network or in the case of tracking problems where the objective is to maintain the system state as close as possible to the desired state within a time interval.

The performance measure must also take into consideration the bounds on admissible controls and states while selecting a reasonable performance measure.

2.0 CONSTRAINED NON-LINEAR OPTIMIZATION

Definition: Constrained minimization is the problem of finding a vector x that is a local minimum to a function f(x) subject to the constraints on the search space of x:

$$\min_{x} f(x)$$

such that one or more of the following holds good:

$$Gi(x) = 0, \quad i = 1,, me$$

$$Gi(x) \leq 0, \quad me + 1, \ldots . m$$

where f(x) is the objective function which returns a scalar and G(x) returns a vector of length m which are the values as a result of evaluation of the constraints at x. Constrained minimization does not aim to find the global optima and uses methods to solve unconstrained optimization within its sub-steps.

2.1 ACTIVE SET ALGORITHM

In constrained optimization, the general aim is to transform the problem into an easier sub-problem that can then be solved and used as the basis of an iterative process. SQP methods represent the state of the art in nonlinear programming methods and within each iteration of the SQP method, a QP problem is solved which produces a solution that is used to form a search direction for a line search procedure.

Given the definition of constrained minimization, the principal idea is the formulation of a QP sub-problem based on a quadratic approximation of the Lagrangian function,

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda i . Gi(x)$$

where λ_i, (i = 1,..m) are Lagrange multipliers. An overview of SQP is found in [1], [2], [3], and [4].

The constrained problem is solved by transforming the constrained problem into a basic unconstrained problem by using a penalty function for constraints that are near the boundary. In this way, the constrained problem is solved by using a sequence of unconstrained optimizations which converge to the constrained problem. The KKT equations are the conditions required for optimality of a constrained problem. The KKT equations are given as,

$$\nabla f(x*) + \sum_{i=1}^{m} \lambda i . Gi(x*) = 0$$

$$\lambda i . Gi(x*) = 0; i = 1,,,,,me$$

$$\lambda i \geq 0; me + 1,,,,,,m$$

3.0 AN OVERVIEW OF NAVIGATION SENSORS

This presents an overview of navigation sensors and how various sensors are used to determine a vehicle's position, ie. Positioning, as well as sensors used for target tracking purposes.

Rangefinders

Rangefinders are among the most popular sensors in autonomous vehicles that is used to measure the range to nearby objects. Range maybe measured along a beam which is a good model of the workings of laser range finders or within a cone which is the preferred model of ultrasonic sensors. A laser is similar to a sonar in the sense that it also actively emits a signal and records its echo using the time-of-flight principle, the only difference being that in the case of laser the signal is light(laser) beam. For very high precision measurements, this technology may not be preferred due to the high speed of light and range free methods such as triangulation is used. This technology is used for surveillance and target tracking by military or law enforcement as a means of calculating distance to the targets to set up a perfect shot.

GPS

The Global Positioning system was conceived as a ranging system using known positions of satellites in space to unknown positions on land, sea, air and space with the objective of instantaneous position and velocity determination for navigation purposes and precise coordination time. It has become the mainstay of transportation systems worldwide and has been used for ground navigation, aviation and maritime operation. GPS uses pseudo-ranges derived from measuring the travel time of the signal and multiplying it by its velocity or measuring the signal's phase. Since the receiving and transmitting clocks are not perfectly synchronized, the pseudoranges obtained account for the clock error. The receiver uses the satellite signal to compute the distance to each satellite by measuring the transit time of the received signal. It then computes its location by using the trilateration technique using the measured distances to 3 satellites. One of the main disadvantages of using GPS for surveying, land navigation is the temporary outage as the receiver passes under obstructions such as bridges or in buildings where GPS is unavailable.

Inertial Navigation Systems

More sophisticated approaches involve combining GPS with Inertial navigation Systems(INS). The inertial navigation system consists of gyros to monitor angular motion of the object's frame with respect to the local frame of reference and accelerometers to measure acceleration. Starting from a known position, double integration of the accelerations over time results differences in position which determine the trajectory of the vehicle. The error of an INS increases with the square of time due to the double integration over time. The advantages of the INS of over GPS are that it is independent of external sources and there is no visibility problem. It also provides accuracy similar to GPS when used over small time intervals and can serve to interpolate measurements during GPS outages.

Pulsed RADAR systems

Radar uses strength and round-trip time measurements of short pulses of electromagnetic waves that are emitted from a radar antenna which are used to determine the range to the target by measuring the time taken for the reception of the reflected signal. The energy in the radar pulse is scattered in all directions and therefore the returned signal is a weak radar echo which is amplified, digitized and processed to display an image.

Continuous Wave(CW) radar systems

As opposed to pulsed radar systems, conventional continuous wave radar systems cannot measure range to the target since electromagnetic signals are transmitted continuously and there is no way of measuring time interval between the transmitted and received signals. The measurements of instantaneous rate of change of the target's range can be measured by the Doppler shift of the returned signal which is the change in frequency of the reflected signal depending on the transmitter's or the target's movement. However, it is also possible to use a CW radar system to measure range instead of range rate by frequency modulation, by measuring the instantaneous difference between the transmitted and received frequencies. This difference is proportional to the time interval between the transmitted and received signals which can be used to determine the target's range as previously discussed.

4.0 RELATED WORK

4.1 MOTION PLANNING APPROACHES

This section gives a brief overview of prior work on navigation in dynamic environments while avoiding obstacles and multi-autonomous vehicle navigation. The problem of collision avoidance has been addressed in several approaches. One such case is of a single autonomous vehicle navigating among moving obstacles by the velocity obstacle approach. This has been employed for various real-world autonomous vehicles such as motion planning in UAV and autonomous vehicles. Variants of velocity obstacle approach has been used to incorporate the mutual response of surrounding obstacles such as recursive probabilistic velocity obstacles, reciprocal velocity obstacles most of which concern with only two robots. Over the recent years there has been an increasing amount of work on autonomous systems operating in a partially observable or a fully observable environment. There has been significant amount of research in the field of probabilistic robotics in the SLAM problem which deals with simultaneous localization of the robot in its environment while learning a map of its environment. However most of the work focuses on working with discrete spaces in discrete time. Most of the existing motion planning techniques that have been implemented are based on graph search methods such as the Djikstra's algorithm, breadth first search and depth first search. Djikstra's algorithm is a form of dynamic programming algorithm which solves the objective function for the shortest path problem using Bellman's principle of optimality.

The potential field approach is another approach to motion planning which considers the robot as a point object in the potential field which combines the effects of attraction to the goal state and repulsion to obstacles. The advantage of this approach is that the trajectory is produced with little computation time. Exact motion planning using complex constraints becomes computationally inefficient.

Motion planning takes into consideration the configuration of the state space. An important consideration is whether the state space is modeled as continuous or discrete. In the discrete case, the state space is a grid world representation where the computation involved in updating the belief depends on the grid resolution. Grid based approaches use a discretized set of actions and algorithms such as A* search are used for used for a priori planning of trajectories. However this is suitable for static environments but requires frequent grid updates for a dynamically changing environment.

Probabilistic planning methods are implemented using probabilistic roadmaps where the roadmap is a graph of paths generated randomly and the waypoints are connected using efficient path planning methods in the configuration space.

The methods discussed including roadmaps, potential fields and cell decomposition come under the category of global planning techniques where the assumption is based on prior availability of the map of the robot's environment. Such approaches are not suitable if a map is unavailable. Global approaches can also be slow because of their complexity. Local approaches assume only a fraction of the environment and therefore are less computationally complex. The vector field histogram approach is implemented which avoids collisions and at the same time steers the mobile robot towards the target. An occupancy grid map is used for representing the robot's environment and this map is continuously updated using ultrasound proximity sensor measurements. Occupancy data is transformed into a histogram representation which aids in computing the motion heading and velocity of the robot. However since vector field histogram(VFH) is a local approach, it does not produce an optimal path since the global information about its environment is unavailable.

4.1.1 Certainty Grid

The certainty grid method was developed by CMU which proposed a probabilistic representation of obstacles. In this application, the robot takes a panoramic scan with its 24 ultrasonic sensors while remaining stationary. Then the robot traverses to the next location, stops and repeats the process of scanning. This process continues until the robot traverses the entire region by which time, the grid records an accurate map of the environment. The grid representation is divided into 2-D elements known as cells where each cell records a certainty value which indicates the probability of existence of an obstacle. These certainty values get updated by a probabilistic sensor model which accounts for the characteristics of sensor used. As an example, the ultrasonic sensor returns a radial distance measurement of the distance to the object that is nearest within its conical field of view. Since the object detected by the sensor is lesser likely to be on the periphery, the probabilistic sensor model increases certainty values along the cells closer to the sensor's acoustic axis. The updation of certainty values of the certainty grid takes place using the probabilistic sensor model, which is applied to the 24 range readings. The advantages of this approach are that it accounts for inaccurate laser range measurements

4.1.2 Virtual Force Field

The Virtual Force Field method is one of the earlier methods developed for fast-vehicles and this method helps in continuous and smooth motion of vehicles amongst obstacles. This application is similar to CMU's certainty grid based approach where each cell on a histogram grid is assigned a certainty value and is continuously updated as sensor measurements come in. However the method by which these certainty values get updated in the virtual force field method is different from that used in the certainty grid approach. The probabilistic sensor model used by the certainty grid approach is applied on all cells that fall within the sensors' readings whereas the Virtual Force Field approach updates only a single cell on the grid for each continuously sampled sensor reading which results in a probability distribution. This results in a probability distribution which results in high certainty values of the cells that are close to the actual location of the obstacle. Finally the potential field concept is applied to the grid where in the vehicle moves over an active region of cells with a preconfigured active window size. For reasons of ease of computation, square windows are used rather than circular windows. For each iteration of the algorithm, the virtual repulsive forces and the virtual attractive forces are summed up to yield a resultant force vector. The repulsive force is proportional to the certainty value of the active cell and inversely proportional to the distance between the cell and the centre of the vehicle. The attractive component of the force is the result of the force of attraction towards the target and is proportional to the distance between the vehicle and the target. The combination of these processes allows the vehicle to quickly react to the presence of obstacle and steer away from it. The disadvantage of this approach is the large number of individual force vectors that have to be summed and is computationally demanding.

As is the case for most of the existing techniques to collision avoidance, the dynamics of the robot motion have not been taken into consideration when obtaining the desired trajectory. This is however an important aspect of safe navigation and has been addressed in curvature velocity approach where the assumption is that the robot moves in circular arcs. The motion commands are obtained using the search space of translational and angular velocities and the dynamics are determined based on a constrained set of admissible velocities. The dynamic window approach to collision avoidance utilizes a space of translational and rotational velocities that obeys a set of preferences and constraints. These constraints are determined by torque limits which in turn limits the space of possible controls. Constraints are also applied given the scenario where a certain velocity would inevitably lead to a collision. A velocity from this space is chosen at regular time intervals. Preferences are represented by an evaluation function which is maximized in order to obtain the best combination of translational and rotational velocity to steer the robot. However, these approaches don’t consider the uncertainties associated with robot motion and sensor characteristics in their kinematic representation and are not specific to a certain problem domain.

4.1.3 Velocity Obstacles

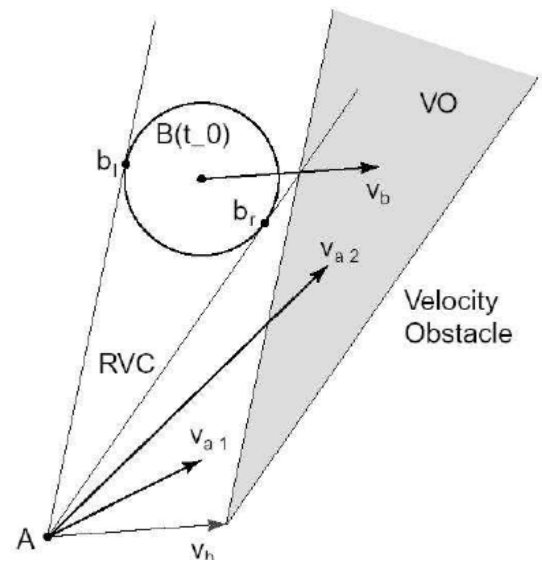


Figure 1. Velocity Obstacle

4.2 MOBILE ROBOT LOCALIZATION AND TRACKING

A multi-target tracking problem is one where the dynamic characteristics of surrounding vehicles are estimated to fulfill objectives depending on the nature of the application. This is achieved by gathering sensor observations on one or more potential obstacles in the environment and to estimate at each step, the position of obstacles' velocities and locations. SONAR was introduced during World War I to be able to track far away objects by American, British and French scientists particularly for submarines and aircrafts along with the development of RADAR technology which was motivated by military objectives.

Filtering techniques to produce estimates of positions, velocities and accelerations were developed driven by applications in image processing and multi-sensor fusion technologies. The Kalman filter is the best studied technique for implementing the Bayes filter for filtering and prediction in linear Gaussian systems. The Kalman filter implements belief computation for continuous states and is not applicable to discrete state spaces. The main issue associated with the multi-target tracking issue in comparison to the single target tracking problem is the target-to-measurement association,. i.e it is necessary to associate an observation gathered by the sensor to each of the existing targets.

4.2.1 Multi-Hypothesis Tracking

A method of multiple-target tracking for associating measurements to targets was the Multi-Hypothesis tracking which represented the belief state by multiple Gaussians. The posterior is expressed by the mixture of Gaussians with each target track corresponding to a mixture component and is an extension of the basic Extended Kalman filter algorithm under unknown data association. This is a disadvantage since the data association problem will be infeasible during situations involving rapid change in number of moving objects in the environment as well as during instances of occlusions. Therefore, the occlusion problem has to be accounted for as a special case and distinctions must be made between stationary and moving targets.

4.2.2 Occupancy Grid Framework

More recently, the occupancy grid framework which was developed by Elfes of CMU was used to enable tracking of multiple moving objects. In this method a cluster of cells are identified and time-stamped based on whether they were occupied on two consecutive timestamps and interpreted as occupied by a stationary object if the condition holds good. This method is efficient for motion tracking and detection than commonly used occupancy grid representations. However, this approach also has the disadvantage of not being able to overcome occlusion effects, since it employs spatio-temporal clustering over temporal maps to perform motion detection and tracking, i.e if a new stationary object is identified in a previously occluded area, the algorithm interprets it as motion in that area.

5.0 PROBLEM DESCRIPTION

5.1 PERFECTLY KNOWN STATES

Problem Formulation

Assumptions:

1) The target's control sequence and state information is perfectly known by the host vehicle does not have uncertainties associated with longitudinal and translational velocities.
2) The host vehicle is currently dealing with two target vehicles and all the vehicles are represented as point objects.
3) The host is constrained to reach a final configuration at a pre-defined time.
4) The initial and final state configurations of the host are specified and the host has perfect knowledge of its states and control information.
5) The motion of vehicles is represented as time-parameterized trajectories

Kinematic motion model

1. Piecewise-constant acceleration

The motion equations below describe robot motion assuming that the robot's translational and rotational velocity can be independently controlled. To realistically evaluate the effects of the control input on the pose vector in a real environment, it is necessary to have complex motion models for a more accurate representation of the resulting pose. But this requires a detailed knowledge of parameters and kinematic configuration that is specific to different types of robots.

As the pose state estimation is carried out on digital hardware, it is necessary to operate in discrete time space and a sampling period is chosen such that the translational and rotational velocities are assumed to be constant within that interval. Under this assumption, the constant acceleration motion model is approximated as a piecewise constant motion and the robot is constrained to move with only a finite number of acceleration commands with a discretized version of the motion model. The general equations of motion are expressed as

$$x(t) = x(t0) + \int_{t0}^{tn} v(t)\cos\big(\theta(t)\big)\,dt$$

$$y(t) = y(t0) + \int_{t0}^{tn} v(t)\sin\big(\theta(t)\big)\,dt$$

where

$$\theta(t) = \theta(t0) + \int_{t0}^{tn} \left\{\omega(t0) + \int_{t0}^{tn} \alpha(t)\,dt\right\} dt$$

x(t) and y(t) represent the robot's position with reference to a global coordinate system and the heading given by θ(t). As mentioned in the case of the constant velocity motion model, the translational velocity v induces forward motion in the heading direction of the robot, and a positive rotational velocity ω creates a left turn. $x(t_0)$ and $y(t_0)$ are the initial coordinates of the robot pose. The velocity v(t) depends on the initial translational velocity $v(t_0)$ and the translational acceleration a(t) within the interval $[t_0 , t]$. The orientation θ(t) depends on the initial heading $\theta(t_0)$, initial rotational velocity $\omega(t_0)$ and the rotational acceleration α(t) within $[t_0 , t]$. If the number of time steps are given by T, then the accelerations a(t) and α(t) are constant within that interval and can be expressed as aΔt and αΔt respectively where Δt corresponds to the interval $[ti, t_{i+1}]$. The resulting motion equations over the points t_0 to t_n in time can be expressed as,

$$x(t) = x(t0) + \sum_{i=1}^{n} \int_{ti}^{ti+1} (v(ti) + a(ti)\Delta t) \cos\left(\theta(ti) + \omega(ti)\Delta t + \frac{1}{2}\alpha(ti)\Delta t^2\right) dt$$

$$y(t) = y(t0) + \sum_{i=1}^{n} \int_{ti}^{ti+1} (v(ti) + a(ti)\Delta t) \sin\left(\theta(ti) + \omega(ti)\Delta t + \frac{1}{2}\alpha(ti)\Delta t^2\right) dt$$

If the vehicle follows a straight line trajectory, $\omega = 0$

If the velocities are kept constant within time intervals, the trajectory of the vehicle can be approximated by piecewise circular segments. The kinematic motion equations are modified to accommodate this case of piecewise constant velocities.

5.2 INITIAL STATE UNCERTAINTY

Problem Formulation:

Assumptions:

1) The target's initial position information is an estimate specified with an associated uncertainty ellipse with 90% confidence.
2) The host vehicle is currently dealing with two target vehicles and all the vehicles are represented as point objects.
3) The target's control sequence is perfectly known and does not have uncertainties associated with longitudinal and translational velocities.
4) The host is constrained to reach a final configuration at a pre-defined time.
5) The initial and final state configurations of the host are specified and the host has perfect knowledge of its states and control information.
6) The motion of vehicles is represented as time-parameterized trajectories

5.3 COMPLETELY UNCERTAIN INFORMATION

Overview

Basic optimal control problems assume perfect information about the robot state and controls at execution. However, in real world applications this assumption is often unrealistic, for example some state variables maybe missing, or the cost of obtaining the exact state values may prohibit the ability to obtain the same. These types of situations are modeled by assuming that the robot receives observations about the value of the current state which has components of stochastic uncertainty. These components include those that are present in the environment model such as geometry, predicted state and current state, uncertainties in the geometry and kinematics of dynamic obstacles as well as in the subject robot. The most important component of uncertainty manifests when the subject robot uses sensors to obtain state information such as its relative distance and bearing to the target obstacles as well as uncertainties due subject as well as target robots. These have a significant impact on the execution of missions and are impossible to discount since they are not small with reference to the tolerance required by the application. The solution to this problem is to use forward simulation methods to estimate the state of the dynamic obstacle by using the knowledge of the initial state estimate and its associated covariance(uncertainty) and controls into a prediction algorithm such as the Extended Kalman Filter and the relative distance and bearing measurements with the subject robot is used for the process of obtaining a posterior distribution of the obstacle state. The algorithm described in the following sections works on the assumption that the subject is equipped with sensors to obtain perfect information about its own state to be able to employ relative localization with target obstacles. By using an error ellipse as desired for the initial state configuration, it is possible to forward simulate the set of possible target trajectories from any given point of time and use this data for the subject robot to obtain an optimal longitudinal and lateral velocity profile as a function of time to achieve its objective.

Problem Formulation

Assumptions:

The assumptions used in the context of optimization under uncertainties are stated as follows,

1) The moving targets are represented as time-parameterized trajectories predicted using an EKF and each associated with an uncertainty distribution.
2) The host is constrained to reach a final configuration at a pre-defined time.
3) The initial and final state configurations of the host vehicle are known and the host has perfect knowledge about its own states and controls.
4) The host vehicle is equipped with rangefinders used to obtain the sequence of range-bearing measurements with the target vehicles over the entire time period at sampling intervals of 0.1s.

EKF-based relative-localization

Bayes Filter – The General problem

The most general algorithm for calculating beliefs is given by the recursive Bayes filter algorithm and this is done by calculating the belief distribution from measurements and control data, that is the belief bel(xt) at time t is calculated from bel(xt-1) at time t-1. The most common form of Bayes filter that is implemented for continuous spaces is the Gaussian filter where the belief states are represented by multivariate Gaussian distributions. The Gaussian filter that is used for linear systems is the Kalman filter and this works on the assumption that observations are linear functions of the state and the next state is a linear function of the previous state, while the non-linear systems use the EKF. The process is executed in two steps:

Prediction step : The algorithm calculates a belief over state xt based on the prior belief over state xt-1 and the control ut and this is obtained by the integral of the product of the two distributions – the prior belief at state xt-1 and the probability of transition from state xt-1 to xt.

Update step : In this step the Bayes filter multiplies the belief bel(xt) by the probability of the measurement obtained, zt. This algorithm is iterated for each possible hypothetical posterior state xt. To obtain an integration to 1, the result is normalized by the normalization constant, which results in the final posterior. This computation is done recursively by considering the initial belief state to be a uniform distribution over the domain of x0 if the initial position x0 is not known with certainty but finite. If the initial position is known exactly, it is initialized as a point mass distribution with the probability mass centered on the exact value of x0,.ie. single realization of the state vector. Partial knowledge of the initial value can be expressed by non-uniform distributions. An update step occurs only if a measurement is available, or else it is omitted.

Probabilistic State Estimation – Illustration

The aim of any localization or tracking problem is to estimate the pose of a robot relative to a global coordinate system using the measurements obtained from control commands and vehicle sensors. Due to the noise in the measurements of real sensors and due to uncertainties in the effect of a control command, the noise needs to be modeled and the resulting estimate of the pose must be determined. The basic process involved is as follows: the robot queries its sensors on its location (eg. GPS, odometry) which returns a Gaussian with a peak that corresponds to the measurement predicted by the sensors, the variance corresponds to the uncertainty in the measurement. Once the measurement has been integrated with the prior belief, the resulting belief has a smaller variance with its mean between the two original means. Once the robot moves towards its right/left, due to the stochastic nature of state transition its uncertainty gets propagated which results in a wider Gaussian that is shifted in the direction and by the amount of robot movement. The resulting posterior is obtained as the robot receives a second measurement with associated uncertainty to update its current belief state.

Kinematic state Representation

The configuration of a mobile robot is commonly represented by six variables, its 3-D Cartesian coordinates and the three Euler angles(roll,pitch,yaw) relative to the external coordinate frame. The pose of a robot in a plane comprises its two-dimensional planar coordinates and its angular orientation described by the following vector which describes its kinematic state configuration:(x y θ)

The orientation of the robot is referred to as the heading direction. The figure below shows the coordinate system for the kinematic model.

Motion Model

To evaluate the effect of the control input on the pose vector of the robot, a motion model is required to predict the future pose based on the current pose and current control input. The motion data ut specifies the velocity commands given to the robot. The two types of

commonly used motion models used for robots operating on a plane are velocity motion model and the odometry motion model. The odometry motion model which gives measures of kinematic information such as distance travelled and angular displacement in practice tends to be more accurate than the velocity motion model, however the odometry information is available only after a control command is executed. For this reason, this type of motion model is not preferred for collision avoidance applications since these types of applications need to predict effects of motion ahead of time. Therefore, here we use predictive models such as velocity motion model and the piecewise constant acceleration model for motion planning.

The motion models that have been used for the robot operating in continuous 2-D plane have been described in the following sections. This part of the implementation uses constant velocity motion model in discrete time by assuming a known noise model for the target robots.

Constant Velocity Motion model (Discrete time)

i. Noise free motion

The velocity motion model assumes that we can control a robot through two velocities, the translational and the rotational velocity. The assumptions are based on positive rotational velocities inducing left turns and positive translational velocities inducing forward motion. For the ideal case of a noise-free robot, the control command at time t is given by the two velocity components as: ut = (vt wt) ^{T}

If both components of the velocity are kept constant for the time interval (t -1, t] , the robot moves along a circular trajectory of radius

$$r = |v/\omega|$$

which follows from the general relationship between translational and rotational velocity, and this equation also includes the case where the robot moves in a straight line in which case $\omega = 0$ and the radius is infinite. Let $x_{t-1} = (x\ y\ \theta)^T$ be the initial pose of the robot, and keeping the velocity constant for the time Δt, the ideal robot will be at $x_t = (x'\ y'\ \theta')^T$ given by

$$\begin{matrix} x' \\ y' \\ \theta' \end{matrix} = \begin{matrix} x \\ y \\ \theta \end{matrix} + \begin{matrix} \frac{-v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t \end{matrix}$$

with respect to the coordinates of the centre of the circle given by

$(xc, yc) = (x - \frac{v}{\omega} sin\theta, y + \frac{v}{\omega} cos\theta)$, explained by the fact that the robot has moved

v. Δt along the circle which caused its heading to change by $\omega\Delta t$.

ii. Probabilistic state transition model for motion under uncertainty

In reality the motion of the robot is subject to noise where the actual control parameters differ from the commanded controls (or measured controls). The motion model is specified by adding samples of motion noise to the true control parameters and the orientation. The resulting velocity is therefore, the commanded velocity added to some Gaussian noise in which the error parameters specify the accuracy of the robot, which indicates lesser accuracy for larger parameters. This is given as follows:

$$\left[\begin{matrix} \tilde{v} \\ \tilde{\omega} \end{matrix}\right] = \left[\begin{matrix} v \\ \acute{\omega} \end{matrix}\right] + \left[\begin{matrix} \acute{\varepsilon}(\alpha 1, \alpha 2) \\ \acute{\varepsilon}(\alpha 3, \alpha 4) \end{matrix}\right]$$

The state transition probability models the effect of the control input ‘ut’ on the robot state xt which is stochastically generated from the state xt-1. Its distribution is expressed by the conditional probability p(xt|xt-1, ut). By enforcing the conditional independence and Markov property we obtain the following relation:

$$P(xt|x0{:}t-1, z1{:}t-1, u, 1{:}t) = p(xt|xt-1, ut)$$

The probability $p(xt|xt-1, ut)$ is obtained with the assumption that the control is carried out for a fixed time interval as specified by the algorithm. The initial part of the algorithm uses the noise free parameters to calculate controls of the robot and the motion error is modeled as the difference between the computed controls and the commanded controls which is described as the motion error. This motion error is modeled as the distribution with zero mean and standard deviation specified by the robot specific-error parameters. The assumption here is based on the independence of different error sources and the desired probabilistic transition model is given by the product of the individual errors as follows:

$$p(xt|xt-1, ut) = prob(v - \tilde{v}, \acute{\varepsilon}(\alpha 1, \alpha 2)) * prob(\acute{\omega} - \tilde{\omega}, \acute{\varepsilon}(\alpha 3, \alpha 4))$$

where $prob(a, b^2)$ models the motion error given by the probability distribution of its parameter *a*’ under a zero-centered normal random variable with variance b^2.

Sensor model

In many robotics applications, features correspond to distinct objects in the environment also known as landmarks and most measurement models are based on the assumption that the position of certain objects in the environment are perfectly to help with localization. The most common model for processing landmarks assume the ability of the sensor to measure the range and bearing measurements relative to the robot's coordinate frame. Landmark measurement models are normally defined for a feature based environment where the map is known ahead and each feature with its known location is identified by a correspondence variable. This location is simply its coordinate in the global reference frame of the map and is modeled as a noise-free landmark whose measurement vector is defined by standard geometric laws. However, this measurement model could also be used in a cooperative collision avoidance scenario where vehicles can communicate poses and estimates amongst each other to achieve pair-wise relative localization.

Hence the measurement model is based on the assumption that the position of the subject robot is assumed to be perfectly known and the positions of moving targets to be estimated using relative localization with host robot's known pose. The current implementation takes into account as the subject robot's environment consisting of moving targets and does not consider stationary obstacles. From the perspective of the target vehicle, its measurement with reference to the subject consists of the laser-range finder measurement (r , phi) where r is the distance to the subject and phi is the bearing angle at which the subject is located within its field of view(The bearing is zero if the subject vehicle is right ahead, that is right in the middle of the target's field of view. If the subject is to the left, the bearing is positive and negative if it is to the right). This gives the model as,

$$zt = h(xt) + N(0, Qt)$$

$$\begin{matrix} r' \\ \Phi' \end{matrix} = \begin{matrix} \sqrt{(\mathrm{xj}-\mathrm{x})^2 + (\mathrm{yj}-\mathrm{y})^2} \\ \mathrm{atan2}(\mathrm{yj}-\mathrm{y}, \mathrm{xj}-\mathrm{x}) \end{matrix} + N(0, Qt)$$

Algorithm details

I. Prediction s tep: The algorithm uses the motion model defined earlier to perform the prediction by adding the noise component to the controls. The equation is restated by substituting true motion controls in place of the commanded controls as follows:

$$\begin{matrix} x' \\ y' \\ \theta' \end{matrix} = \begin{matrix} x \\ y \\ \theta \end{matrix} + \begin{matrix} \frac{-\tilde{v}}{\acute{\omega}}\sin\theta + \frac{\tilde{v}}{\acute{\omega}}\sin(\theta + \acute{\omega}\Delta t) \\ \frac{\tilde{v}}{\acute{\omega}}\cos\theta - \frac{\tilde{v}}{\acute{\omega}}\cos(\theta + \acute{\omega}\Delta t) \\ \acute{\omega}\Delta t \end{matrix}$$

The prediction is performed as,

$$\overline{bel(xt)} = \int p(xt \mid xt-1, ut)\ bel(xt-1)\ dxt$$

Here xt-1 and xt are the state vectors at time t-1 and t respectively and the true motion is specified by the translational and rotational velocity components generated by the control ut with Gaussian noise.

$$\begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \acute{\omega} \end{bmatrix} + \begin{bmatrix} \acute{\varepsilon}(\alpha 1, \alpha 2) \\ \acute{\varepsilon}(\alpha 3, \alpha 4) \end{bmatrix}$$

Linear transformation using Taylor expansion: State transitions and measurement models in realistic robotics applications are rarely linear and motion models account for circular trajectories which cannot be expressed as linear state transitions. The EKF is designed to handle non-linear state transition and observation models and separates the noise component from the motion model as,

$$\begin{matrix} x' \\ y' \\ \theta' \end{matrix} = \begin{matrix} x \\ y \\ \theta \end{matrix} + \begin{matrix} \frac{-\mathrm{v}}{\omega}\sin\theta + \frac{\mathrm{v}}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{\mathrm{v}}{\omega}\cos\theta - \frac{\mathrm{v}}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t \end{matrix} + N(0, Qt)$$

EKF linearization approximates the state transition function through a Taylor expansion:

$$g(ut, xt-1) = g(ut, xt-1) + Gt(xt-1 - \mu t - 1)$$

The linearization process forms the linear approximation to the non-linear function by a linear function from the former’s value and slope at the mean of the posterior u. This is because the exact state $xt-1$ is not known, and is substituted by its expected value $\mu t-1$

which is known. The advantage of this process results in a closed form computation of the resulting Gaussian. The slope is given by the partial derivative of the transition function as

$$Gt = \frac{\partial g(ut,\mu t-1)}{\partial xt-1} = \begin{matrix} \frac{\partial x'}{\partial \mu x,t-1} & \frac{\partial x'}{\partial \mu y,t-1} & \frac{\partial x'}{\partial \mu \theta,t-1} \\ \frac{\partial y'}{\partial \mu x,t-1} & \frac{\partial y'}{\partial \mu y,t-1} & \frac{\partial y'}{\partial \mu \theta,t-1} \\ \frac{\partial \theta'}{\partial \mu x,t-1} & \frac{\partial \theta'}{\partial \mu y,t-1} & \frac{\partial \theta'}{\partial \mu \theta,t-1} \end{matrix}$$

$$Gt = \begin{matrix} 1 & 0 & \frac{\mathrm{vt}}{\omega\mathrm{t}}(-\cos(\mu\theta,t-1)+\frac{\mathrm{vt}}{\omega\mathrm{t}}\cos(\mu\theta,t-1+\ \omega\Delta\mathrm{t})) \\ 0 & 1 & \frac{\mathrm{vt}}{\omega\mathrm{t}}(-\sin(\mu\theta,t-1)+\frac{\mathrm{vt}}{\omega\mathrm{t}}\sin(\mu\theta,t-1+\ \omega\Delta\mathrm{t})) \\ 0 & 0 & 1 \end{matrix}$$

where $(\mu x,t-1\ \ \mu y,t-1\ \ \mu\theta,t-1)$ denotes the mean estimate factored into the three

dimensions. To determine the covariance of the motion noise, it is necessary to map the motion noise from the control space to state space using another linear transformation which is obtained as the derivative of the motion function with respect to the motion parameters.

$$Vt = \frac{\partial g(ut,\mu t-1)}{\partial ut} = \begin{matrix} \frac{\partial x'}{\partial vt} & \frac{\partial x'}{\partial \omega t} \\ \frac{\partial y'}{\partial vt} & \frac{\partial y'}{\partial \omega t} \\ \frac{\partial \theta'}{\partial vt} & \frac{\partial \theta'}{\partial \omega t} \end{matrix}$$

$$Vt = \begin{matrix} \frac{-\sin\theta + \sin(\theta + \omega t\Delta t)}{\omega t} & \frac{vt}{\omega t^2}(\sin\theta - \sin(\theta + \omega t\Delta t) + \frac{vt}{\omega t}\cos(\theta + \omega\Delta t)\Delta t \\ \frac{-\cos\theta - \cos(\theta + \omega t\Delta t)}{\omega t} & -\frac{vt}{\omega t^2}(\cos\theta - \cos(\theta + \omega t\Delta t) + \frac{vt}{\omega t}\sin(\theta + \omega\Delta t)\Delta t \\ 0 & \Delta t \end{matrix}$$

The covariance consists of two components; one estimating the uncertainty due to the location and the other due to motion noise as,

$$\Sigma t = G_t \Sigma_{t-1} G_t + V_t M_t V_t^T$$

II. Update step : To perform the update step, the algorithm also requires a linearized range-bearing measurement model with additive Gaussian noise at the predicted mean $\bar{\mu}_t$ as,

$$h(xt) = h(\bar{\mu}t) + Ht(xt - \bar{\mu}t)$$

Where,

$$Ht = \frac{\partial h(\bar{\mu}_t)}{\partial xt} = \begin{matrix} \frac{\partial r'}{\partial \mu x,t} & \frac{\partial r'}{\partial \mu y,t} & \frac{\partial r'}{\partial \mu\theta,t} \\ \frac{\partial \Phi'}{\partial \mu x,t} & \frac{\partial \Phi'}{\partial \mu y,t} & \frac{\partial \Phi'}{\partial \mu\theta,t} \end{matrix}$$

$$Ht = \begin{matrix} -\frac{xj - x}{\sqrt{d}} & -\frac{yj - y}{\sqrt{d}} & 0 \\ \frac{yj - y}{d} & -\frac{xj - x}{d} & -1 \end{matrix}$$

where d = $(xj - x)^2 + (yj - y)^2$

Measurement Likelihood

Similar to the state transition model, the stochastic measurement model can be expressed as a conditional probability

⇨ $p(zt \mid z1{:}t-1, u1{:}t, xt) = \int p(zt \mid xt, z1{:}t-1,\ u1{:}t)\ p(xt \mid z1{:}t-1,\ u1{:}t)\ dxt$

⇨ $= \int p(zt \mid xt)\ p(xt \mid z1{:}t-1,\ u1{:}t)\ dxt$

⇨ $= \int p(zt \mid xt)\ \overline{bel(xt)}\ dxt$

The EKF algorithm computes the likelihood as a convolution of two Gaussians, one representing the state uncertainty and the other representing the measurement uncertainty and after the linear approximation gives the likelihood function,

$p(zt \mid z1{:}t-1, u1{:}t, xt)$ = N (zt; $h(\bar{\mu}t)$ + H_t $(xt - \bar{\mu}t)$, Q_t) ⊗ N(x_t , $\bar{\mu}t, \overline{\Sigma t}$)

= N (zt; $h(\bar{\mu}t)$, $H_t \overline{\Sigma t} H_t^T + Q_t$)

6.0 EXPERIMENTAL SETUP AND RESULTS

6.1 APPLICATION TO SPECIFIC CASES AND SCENARIOS

Scenario-1: Suppose we consider a situation where the host vehicle wants to reach a goal state configuration specified by a vector that is straight ahead along its path and target vehicle 1 makes change into the host vehicle's path. The host vehicle follows a curvilinear trajectory and turns back into its original path. The target's initial position and control sequence is known by the host vehicle. The weights used in a particular instance are : $k1 = 1, k2 = 10 \text{ and } k3 = 0.01$.

Scenario-2 : The host vehicle optimizes with 2 targets instead of 1 with all vehicles following curvilinear paths crossing each other.

The matlab function *fmincon* is used to solve the non-linear constrained optimization problem. The initial parameters and inequality constraints supplied to the *fmincon* function are as follows and are applicable to all scenarios described:

i. **Initial parameter vector $\boldsymbol{x}$**: $[t1\ t2\ vmax\ wmax]$ and the corresponding values are $[3\ 4\ 20\ 2]$

ii. **Vector of lower bounds** : $lb = [0\ 0\ 0\ 0]$

iii. **Vector of upper bounds** : $ub = [5\ 10\ 50\ 100]$

iv. **Matrix for inequality constraints** : $Aineq = [2\ 1\ 0\ 0]$; $Bineq = 10$

The motion profiles of the participating vehicles are given by describing their longitudinal and translational velocities over the time interval from the initial time to the time of arrival. The following section describes the parameterization.

6.2 PARAMETERIZATION FOR TRAJECTORY PLANNING

The translational and rotational velocity profiles are specified according to nature of motion of the autonomous vehicles. Depending upon priorities desired by the host vehicle, parameters and weights are chosen to formulate the objective function.

The parametric forms for the longitudinal and angular velocity profiles of all the participating vehicles are as shown below. The trajectory profiles follow as a result of the parametric forms and are illustrated for individual vehicles in the set of subplots.

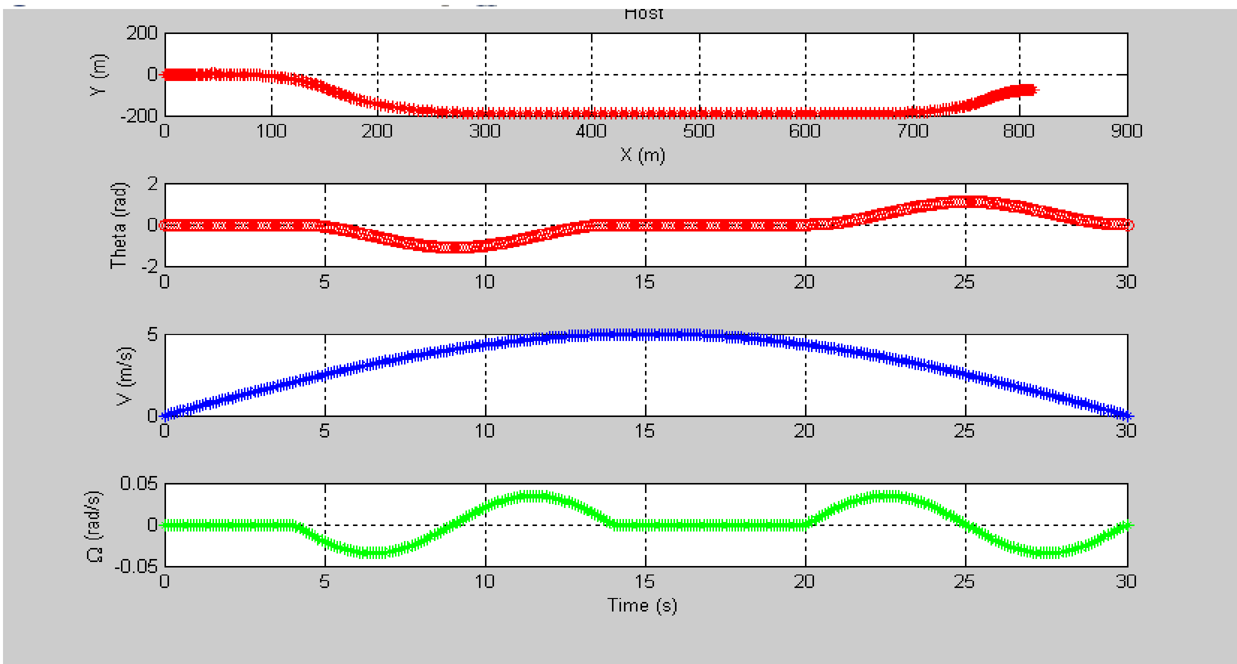


Figure 2. Host parameters

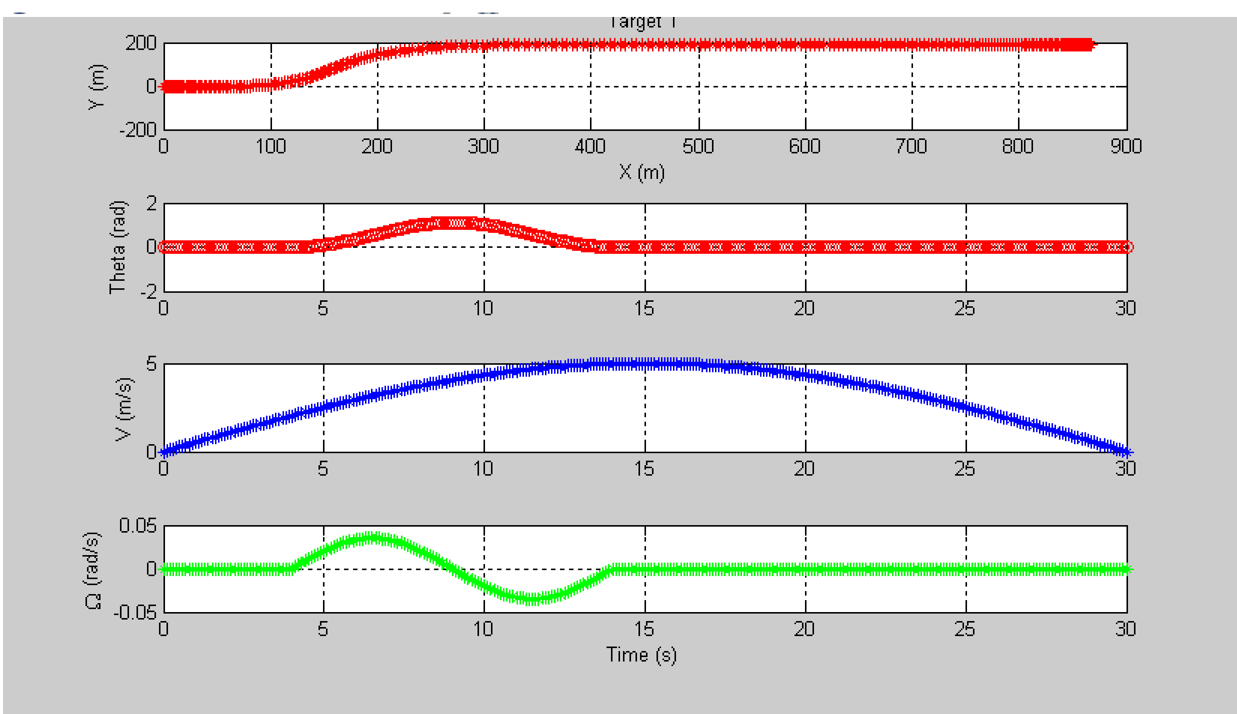


Figure 3. Target 1 parameters

6.3 TRAJECTORY OPTIMIZATION

The goal of the objective function with respect to the host vehicle is to reach the goal state configuration within a specified time Tf and is optimized for the maximum attainable translational and rotational velocities, and the time periods t1 and t2 corresponding to intervals of straight line motion. The objective function to be minimized includes the following criteria:

1) $dist(h,k)$:The minimum distance to the target vehicles given by $dist(h,k)$ which is a function that determines the minimum of the distance vector between the target and host vehicles over the entire trajectories. The aim of the host vehicle is to maximize this distance to each of the targets.

2) $dist(h, final)$: Distance to the end state must be minimized to reach within the time constraint, Tf.

3) $wmaxhost$: Maximum angular velocity that the host must use to successfully change trajectory.

$$J = k1.dist(h, final) + k2.\frac{1}{\sum_{k=0}^{n} dist(h,k)} + k3.1/\omega\text{maxhost}$$

where n is the number of target vehicles.

This minimization problem is solved using the active-set method which is a non-linear optimization technique that uses a set of constraints to determine the effect each of the constraints would have on the objective function. The objective function uses the following bounds and inequality constraints:

i. $2 * t1 + t2 \leq T$

ii. t1 $\in$ [lb ub]

iii. t2 $\in$ [lb ub]

iv. ωmax $\in$ [lb ub]

where T is the time period of the sine function of angular velocity corresponding to a trajectory change and t1 and t2 correspond to the first and second intervals of straight line motion in between trajectory changes where t1 $\in$ [lb ub] , t2 $\in$ [lb ub] and ωmax $\in$ [lb ub]. The constants k1, k2 and k3 are used as weighting factors for the three criteria used in the objective function.

The combination of the three factors is varied depending upon the priorities required by the objective and the results of varying the weights with different scenario are illustrated in detail later.

The same objective function and constraints are used across all three cases with minor modifications. In case of uncertain initial position, the optimization is performed for the set of trajectories starting with the initial set of points within the uncertainty ellipse. In the case of uncertainty in motion, states and measurements, the objective function includes the negative log-likelihood of the measurement probability given the target positions,

$$J = k1.dist(h, final) + k2.\frac{1}{\sum_{k=0}^{n} dist(h,k)} + k3.\frac{1}{\omega \text{maxhost}} - \text{k4}.\log(p(zt \mid z1{:}t - 1, u1{:}t, xt))$$

7.0 SIMULATION AND OPTIMIZATION RESULTS

Once the parametric forms of the velocities and initial optimization parameters are specified, the simulation is performed and if the gradients are not provided, fmincon computes its own numerical approximations to the gradient. The inequality constraints of the system are provided to the function and active-set algorithm is chosen as the optimization algorithm option for this problem. The algorithm iterates until convergence to the optimal values. The simulation of the optimal trajectory of the host vehicle using the optimal values of $vmax$ and $wmax$ for the various scenarios and problem assumptions are shown in the subsequent sections.

7.1 OPTIMAL CONTROL UNDER PERFECTLY KNOWN STATE INFORMATION

Scenario 1: Simulation results after obtaining the optimal parameters as :

t1 = 3, t2= 4, v =18.3563 ω =1.9277

Initial parameters : t1 = 3, t2= 4, v =20 ω =2

Weighting factors : k1 =1, k2 = 10, k3 = 0.01

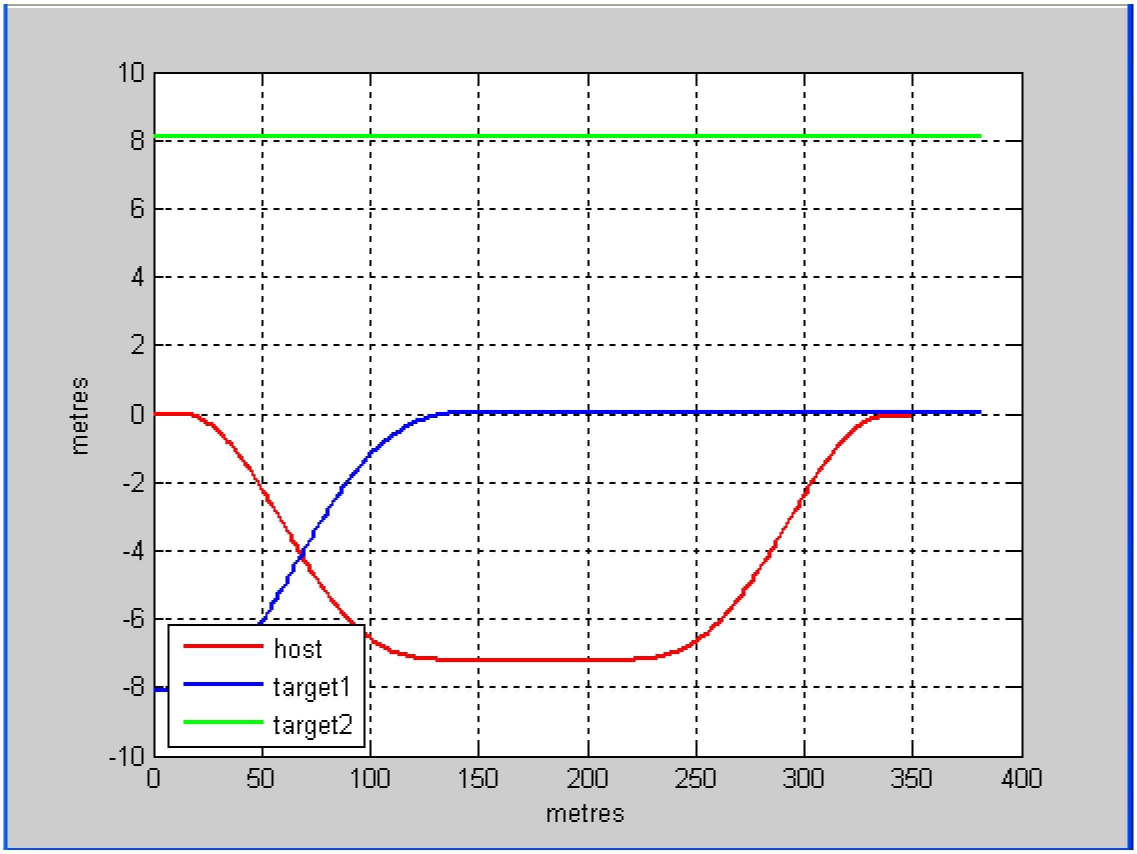


Figure 4. Scenario 1 - known states

The change in the velocity profiles at each iteration of the algorithm until the optima are reached are shown as follows,

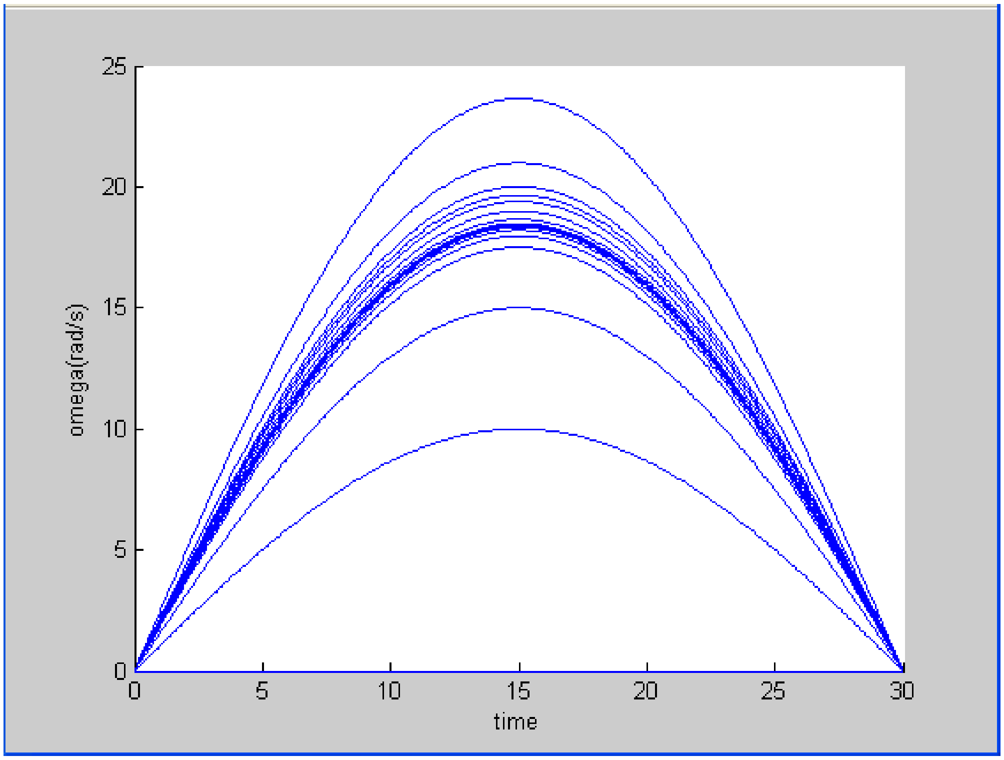


Figure 5. Time(s) vs. V(m/s)

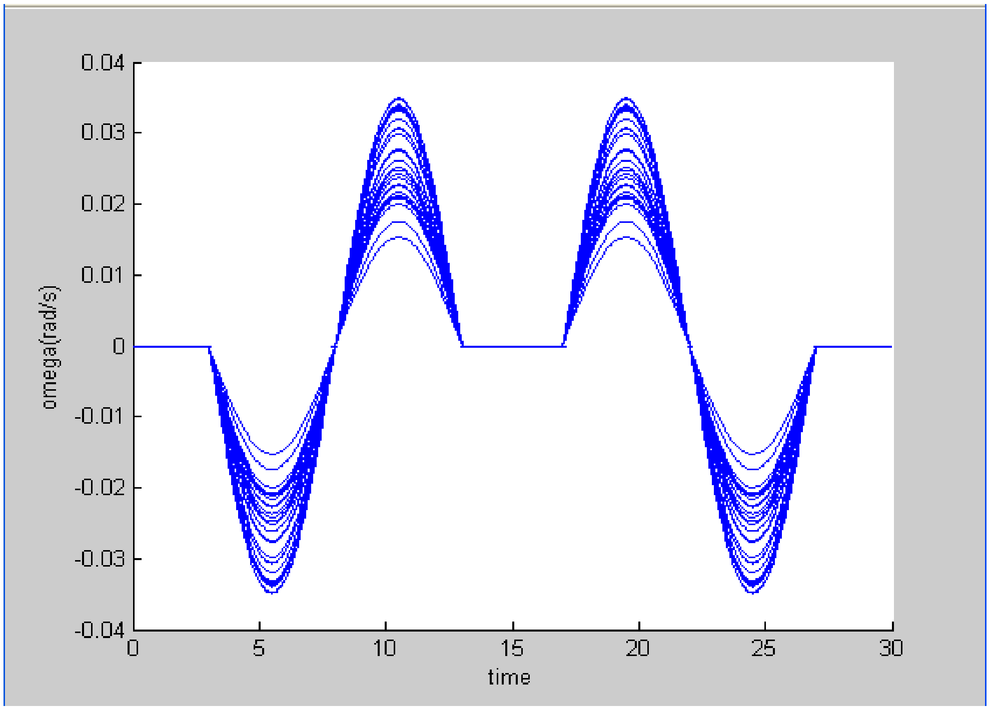


Figure 6. Time(s) vs. ω(rad/s)

Simulation results after obtaining the optimal parameters as :

t1 = 3, t2= 4, v =18.3547 ω =1.8755

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 10, k3 = 0.01

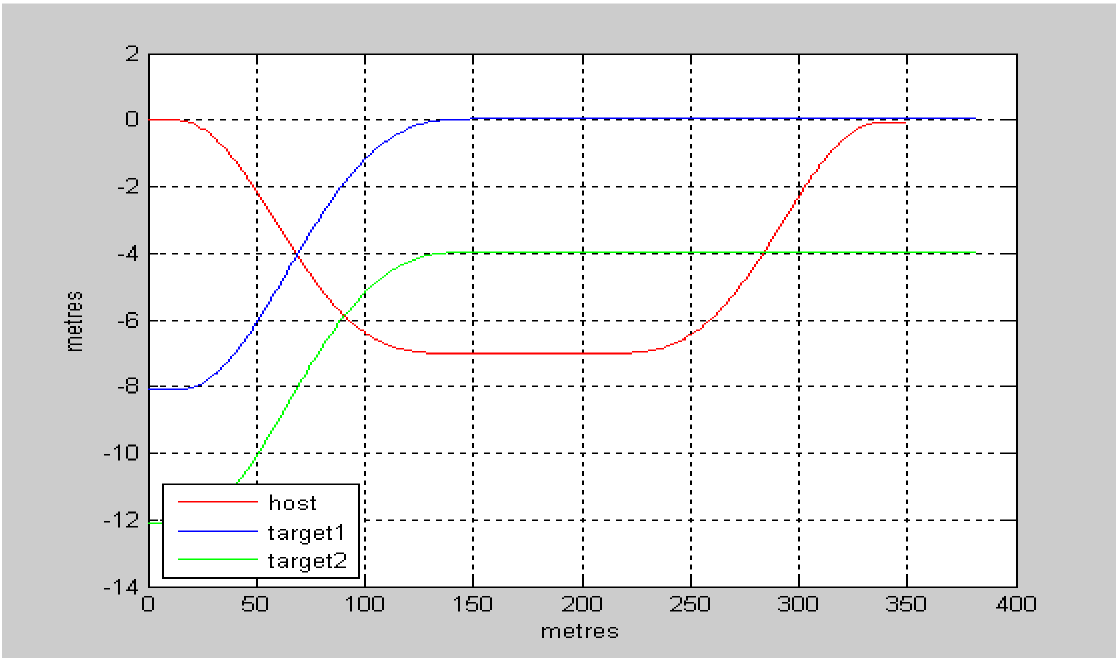


Figure 7. Simulation results

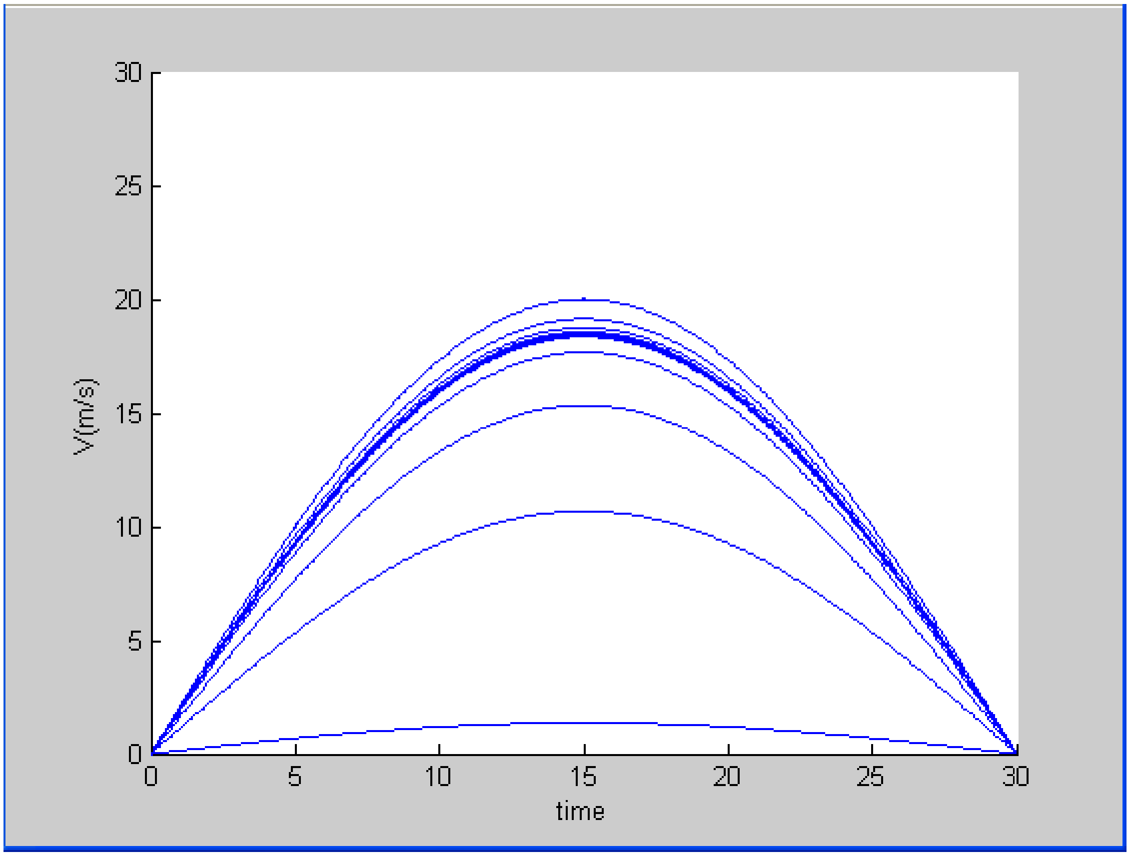


Figure 8. Time(s) vs. V(m/s)

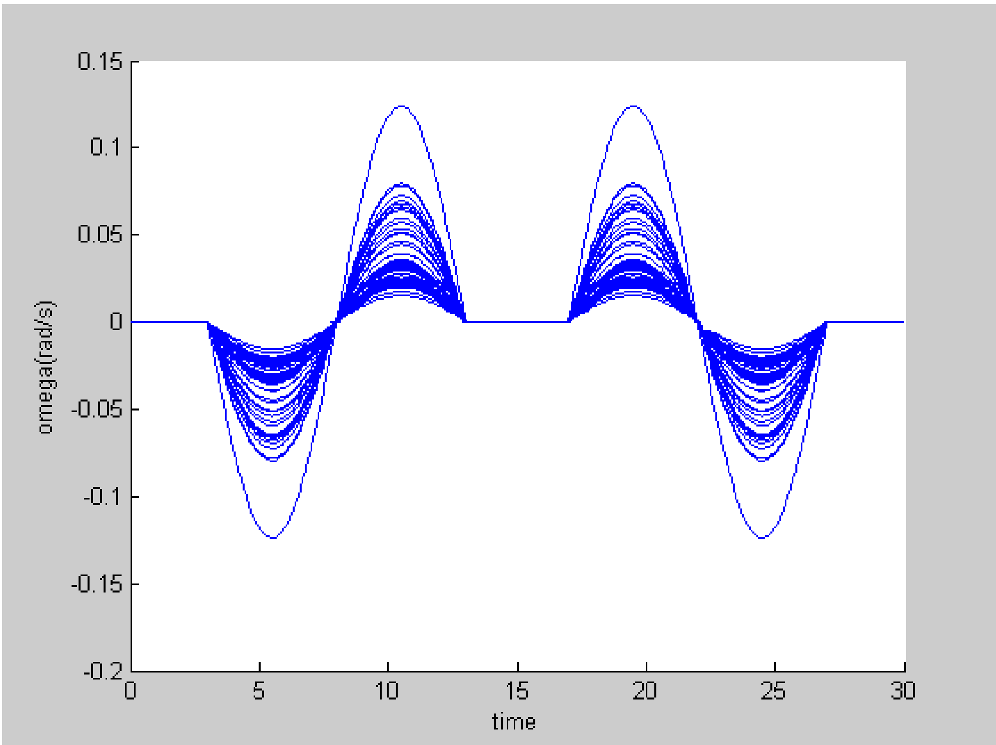
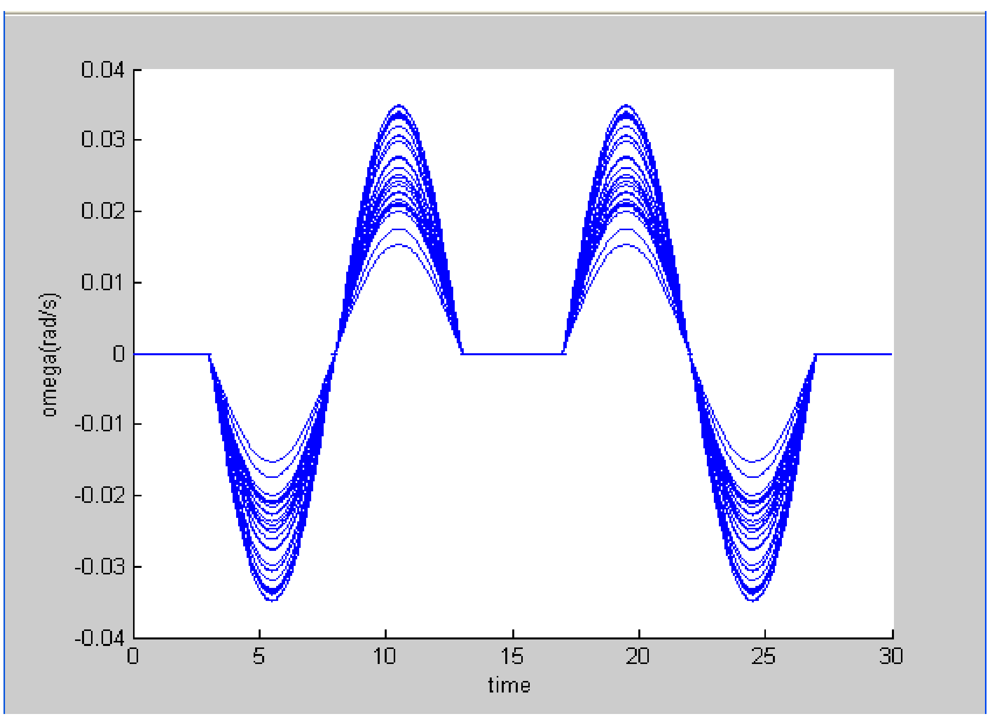


Figure 9. Time(s) vs. ω(rad/s)



Simulation results after obtaining the optimal parameters as :
t1 = 3, t2= 4, v =18.3547 ω =1.8755

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 10, k3 = 0.01

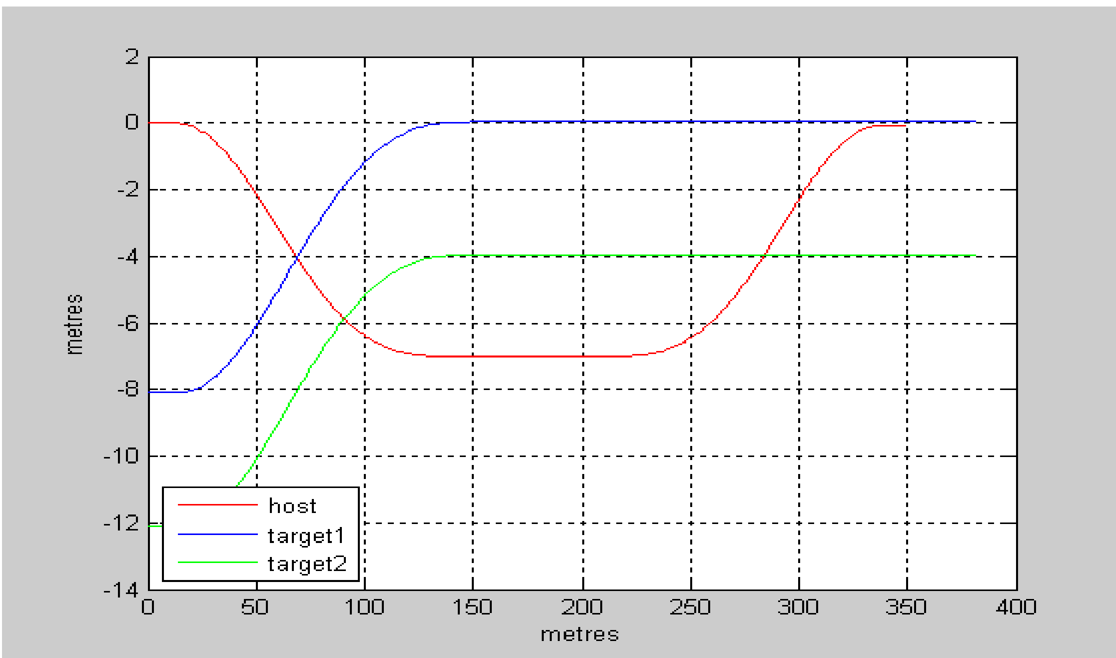


Figure 10. Simulation results

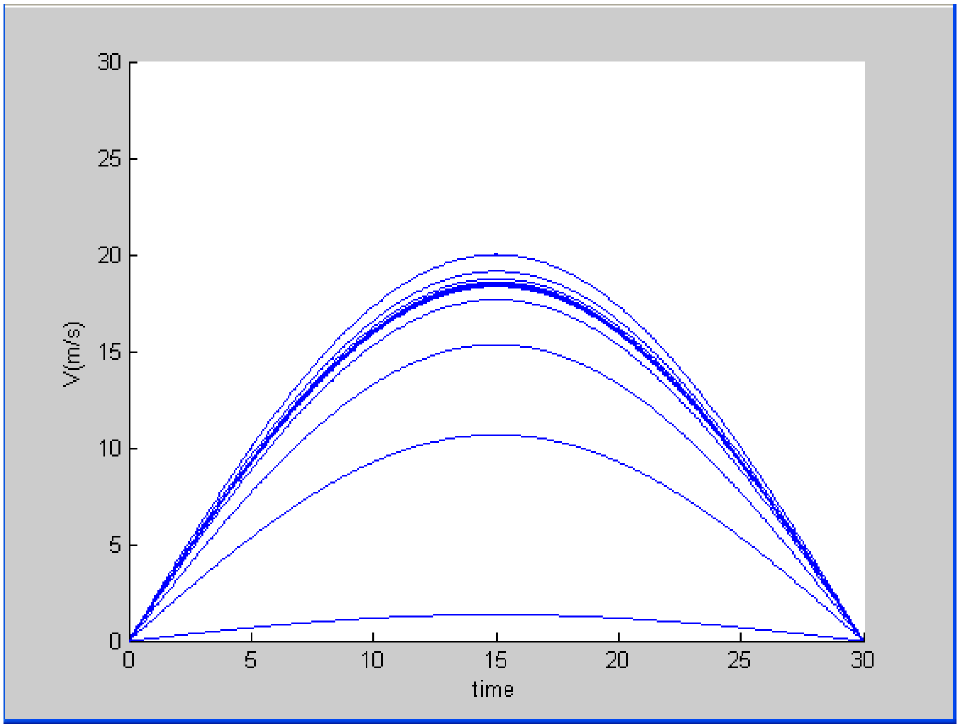


Figure 11. Time(s) vs. V(m/s)

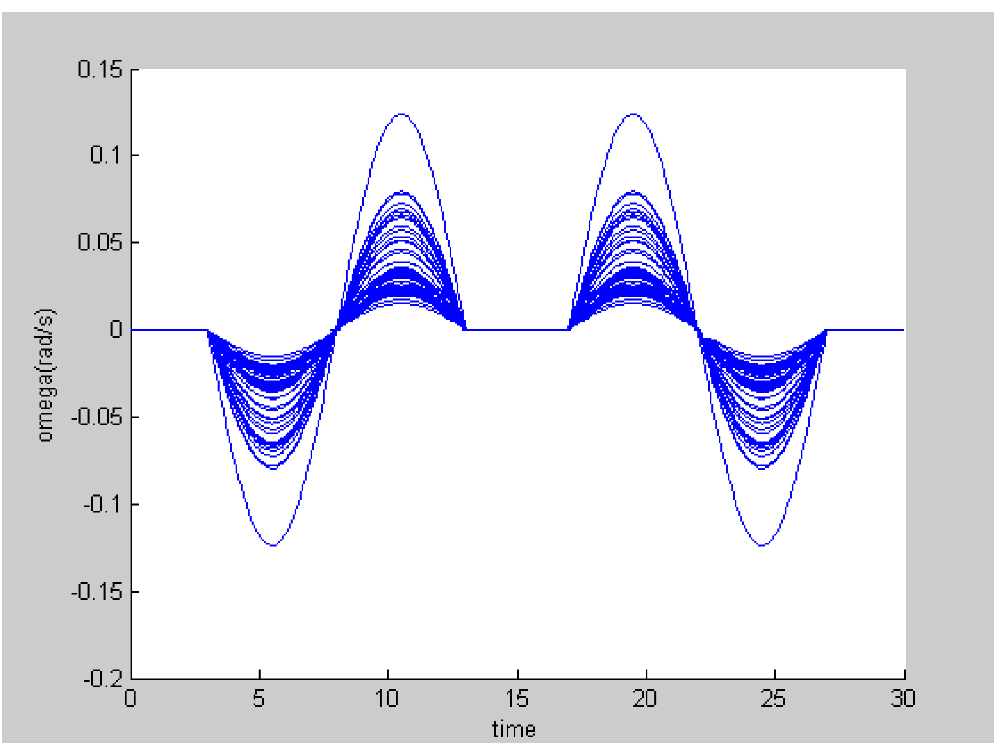


Figure 12. Time(s) vs. ω(rad/s)

7.2 OPTIMAL CONTROL UNDER UNCERTAINTY IN THE TARGET'S INITIAL POSITION

Scenario 1: Simulation results after obtaining the optimal parameters as :
t1 = 2.6547, t2= 3.1369, v =18.3164 ω =1.7541

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 10, k3 = 0.01

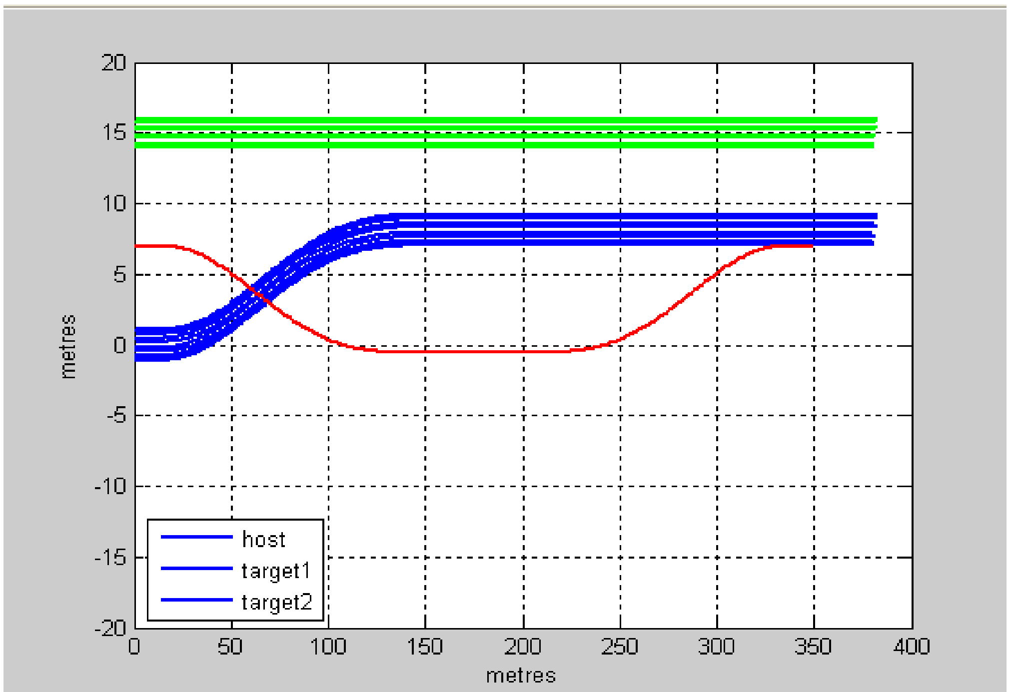


Figure 13. Simulation results- uncertain initial state

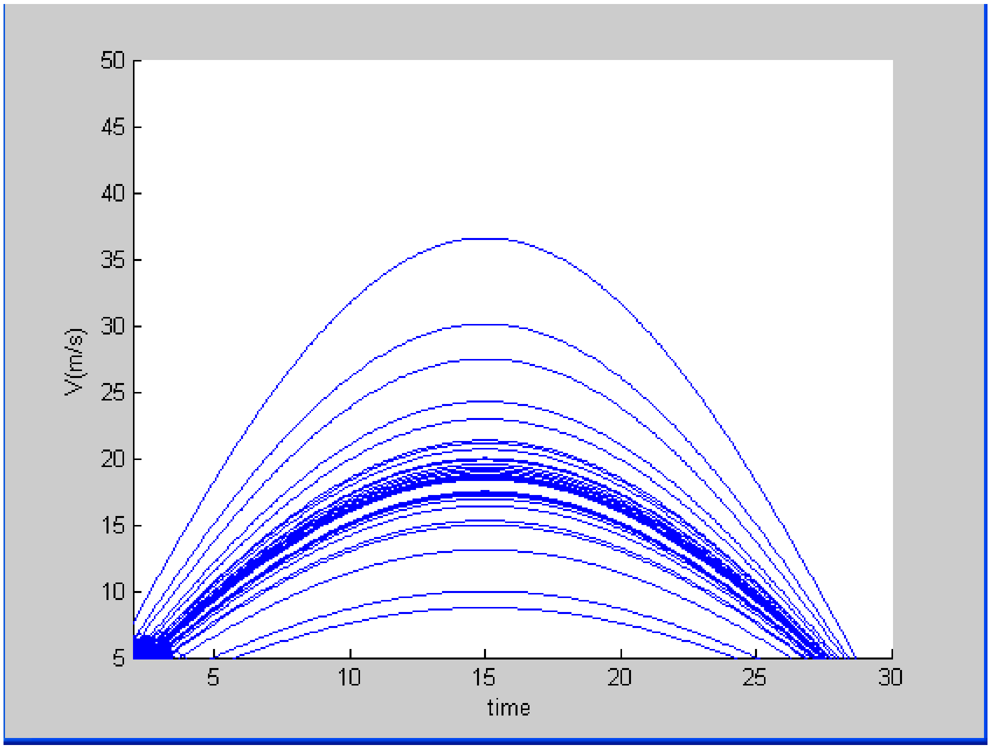


Figure 14. Time(s) vs. V(m/s)

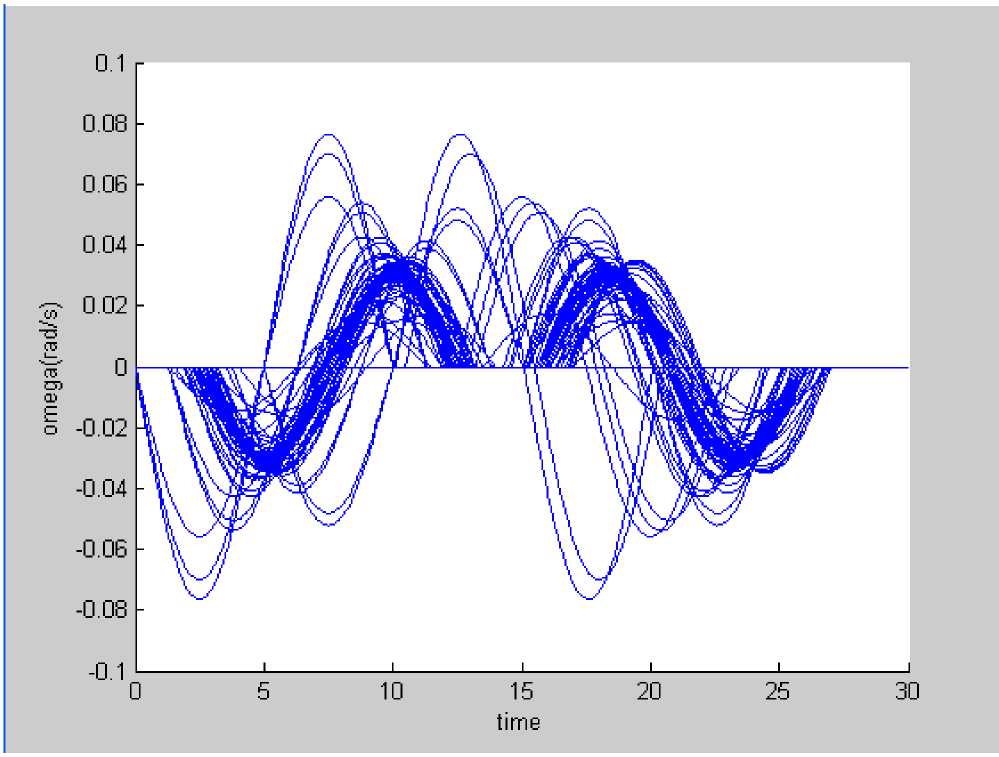


Figure 15. Time(s) vs. ω(rad/s)

Scenario 2: Simulation results after obtaining the optimal parameters as :
t1 = 1.6853, t2= 2.0318, v =18.1153 ω =1.0434

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 10, k3 = 0.01

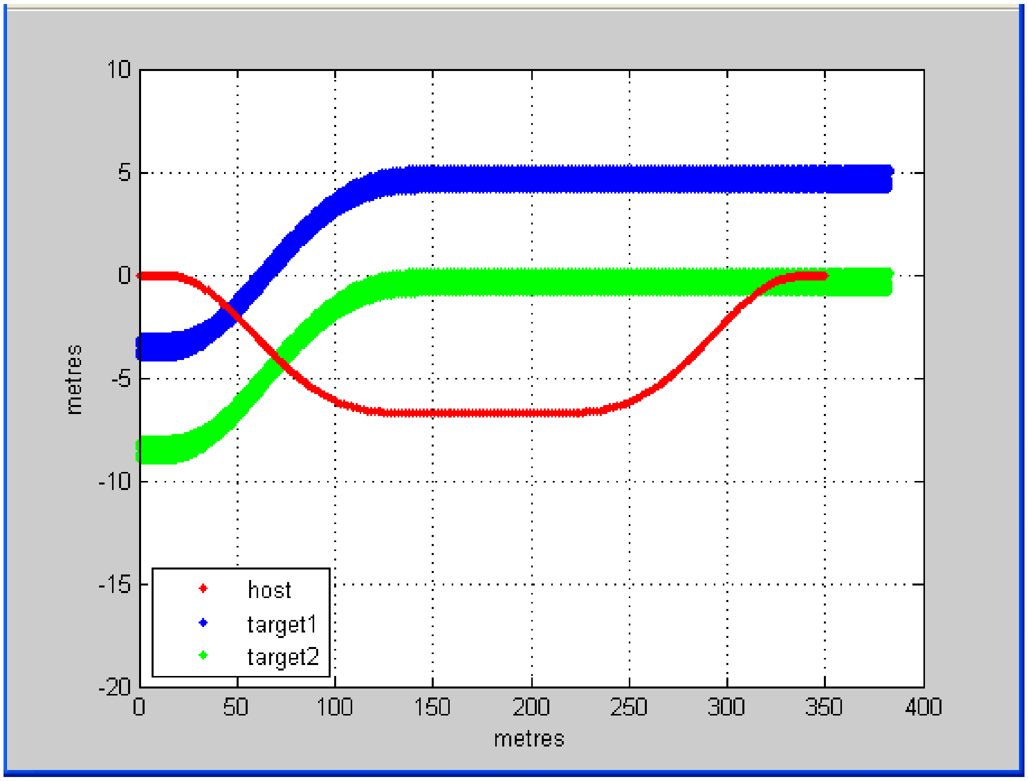


Figure 16. Simulation results

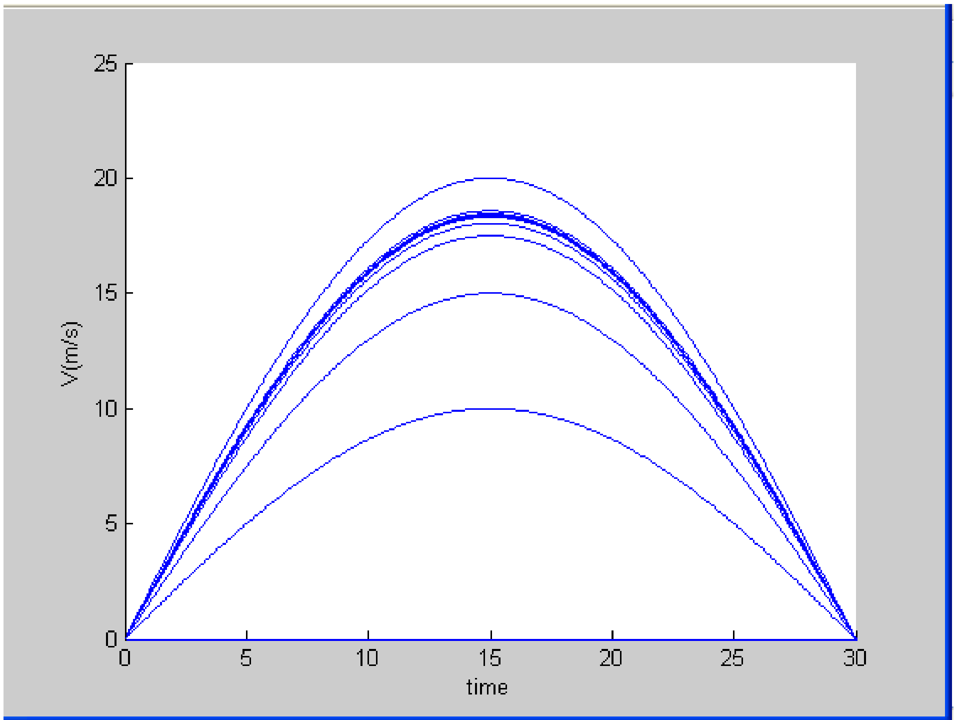


Figure 17. Time(s) vs. V(m/s)

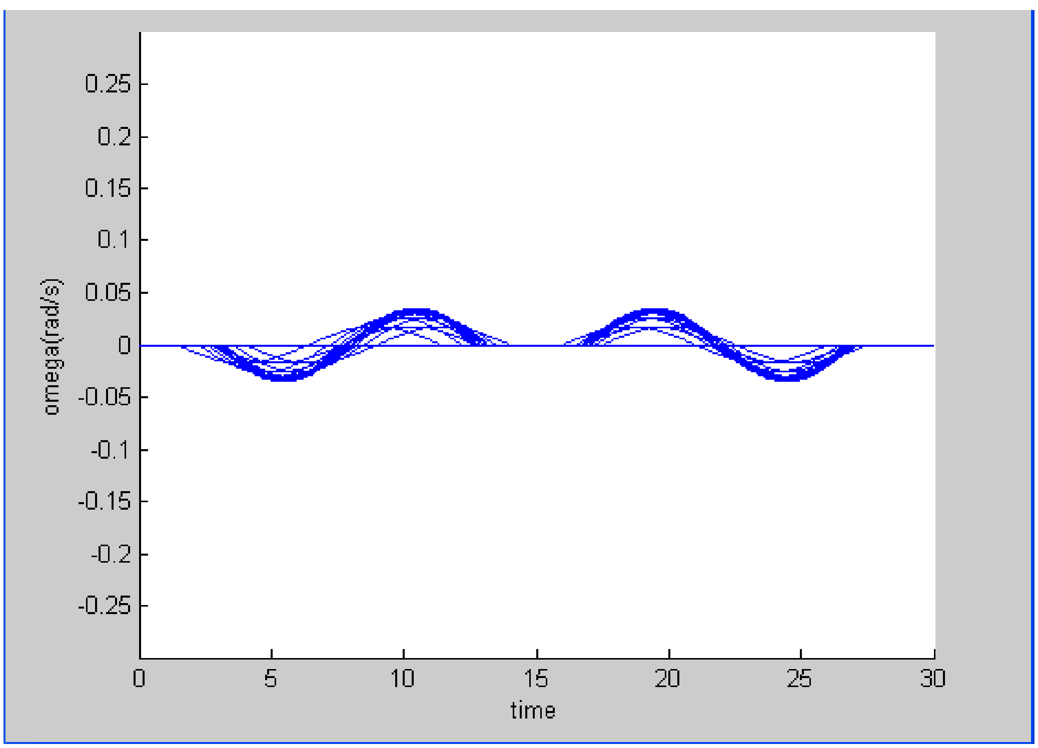


Figure 18. Time(s) vs. ω(rad/s)

7.3 OPTIMAL CONTROL UNDER STATE, MOTION AND MEASUREMENT UNCERTAINTY

Scenario 1: Simulation results after obtaining the optimal parameters as :
t1 = 2.6247, t2= 4.7495, v =23.7473 ω =1.7502

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 1, k3 = 0.01, k4=1

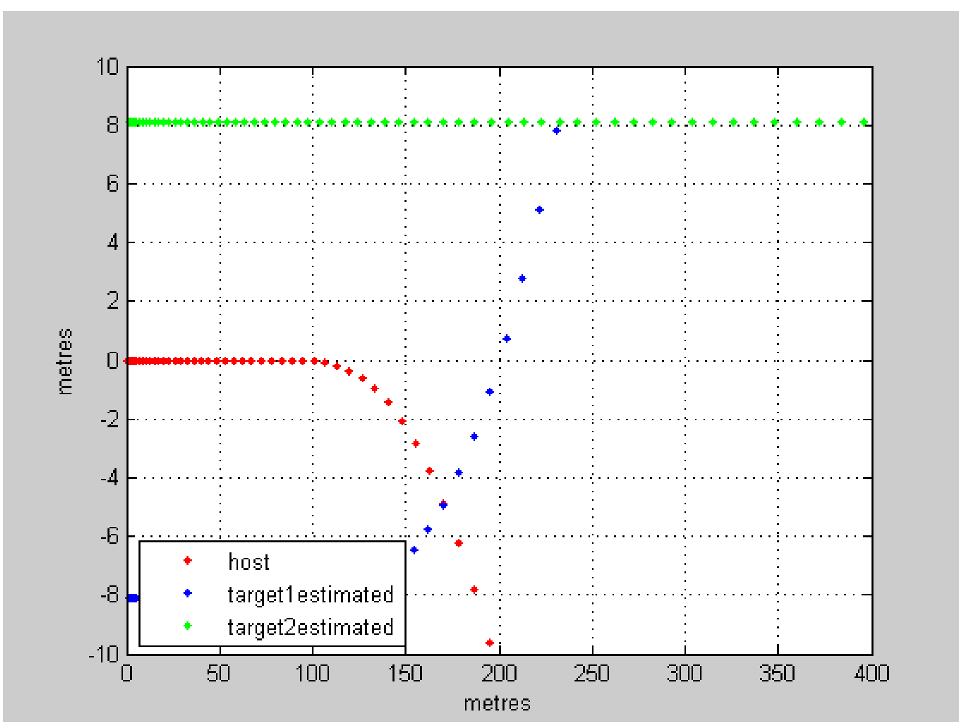


Figure 19. Simulation results: Scenario 1- Uncertain information

Scenario 2 : Simulation results after obtaining the optimal parameters as : t1 = 3.2502, t2= 3.3526, v =26.5797 ω =2.0679

Initial parameters: t1 = 3, t2= 4, v =20 ω =2

Weighting factors: k1 =1, k2 = 1, k3 = 0.01, k4=1

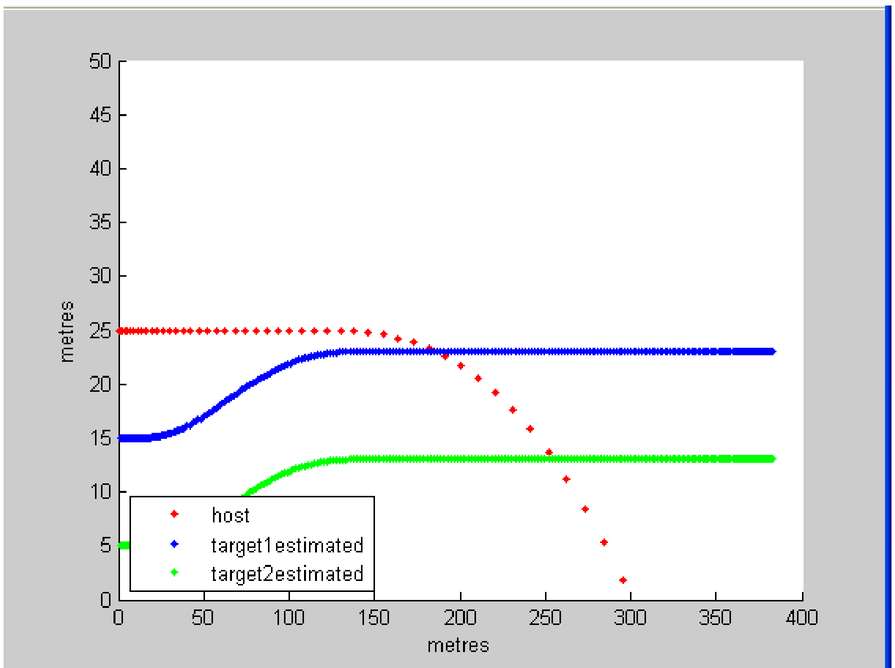


Figure 20. Simulation results: Scenario 2- Uncertain information

BIBLIOGRAPHY

[1] “Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)”, Sebastian Thrun, Wolfram Burgard, and Dieter Fox

[2] "Practical Methods of Optimization", Fletcher, R, John Wiley and Sons, 1987

[3] “Practical Optimization”, Gill, P.E., W. Murray, and M.H.Wright, London, Academic Press, 1981

[4] "Variable Metric Methods for Constrained Optimization", *Mathematical Programming: The State of the Art*, (A. Bachem, M. Grotschel and B. Korte, eds.) Springer Verlag, pp 288-311, 1983

[5] “Map-Based Precision Vehicle Localization in Urban Environments” Jesse Levinson, Michael Montemerlo, Sebastian Thrun Stanford Artificial Intelligence Laboratory

[6] “ Simultaneous Localization and Mapping “ , Pouyan TaghipourBibalan

[7] "A Tesselated Probabilistic Representation for Spatial Robot Perception and Navigation", Alberto Elfes

[8] "Kalman Filter Based Integration of DGPS and Vehicle Sensors for Localization", Shahram Rezaei, Member, IEEE, Raja Sengupta, Senior Member, IEEE

[9] "Including probabilistic target detection attributes into map representations", John Mullane, Ebi Jose, Martin D. Adams, Wijerupage Sardha Wijesoma

[10] "A Path Tracking Method For Autonomous Mobile Robots Based On Grid Decomposition" A. Pozo-Ruz*, C. Urdiales, A. Bandera, E. J. Pérez and F. Sandoval

[11] "Real-Time Computation of Strategies of Differential Games with Applications To Collision Avoidance", Lachner, R.,* Breitner, M.H., and Pesch, H.J

[12] "Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid" , Chiara Fulgenzi, Anne Spalanzani, and Christian Laugier
Laboratoire d'Informatique de Grenoble, INRIA Rhone-Alpes, France

[13] "Localization in Occupancy Grid Maps Using Sonar", http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/fox99a-html/node23.html

[14] "Path planning for sensing multiple targets from an aircraft", Masters' thesis by Jason K. Howlett

[15] “Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenario”, Javier Minguez, Associate Member, IEEE, and Luis Montano, Member, IEEE

[16] "Trajectory Generation and Path Planning for Autonomous Aerobots" , S. Sharma, E.A. Kulczycki, A. Elfes

[17] "A Probabilistic Technique for Simultaneous Localization and Door State Estimation with Mobile Robots in Dynamic Environments" , Dzintars Avots1, Edward Lim1, Romain Thibaux, Sebastian Thrun

[18] "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots", Johann Borenstein, Member, IEEE, and Yoram Koren, Senior Member, IEEE

[19] "Range-Free Localization", Radu Stoleru, Tian He and John A. Stankovic