

Linear Time Varying MPC Based Path Planning of an Autonomous Vehicle via Convex Optimization

Mohsen Ahmadi Mousavi¹

Behzad Moshiri²

Zainabolahoda Heshmati³

¹Graduate Student of Mechatronics Engineering, Faculty of New Sciences and Technologies,
University of Tehran, Tehran, Iran
s.m.a.mousavi@ut.ac.ir

²Professor, Control and Intelligent Processing Center of Excellence (CIPCE), College of Engineering,
School of Electrical and Computer Engineering,
University of Tehran, Tehran, Iran
moshiri@ut.ac.ir

³Assistant Professor, Faculty of New Sciences and Technologies,
University of Tehran, Tehran, Iran
zheshmati@ut.ac.ir

Abstract :

In this paper a new method is introduced for path planning of an autonomous vehicle. In this method, the environment is considered cluttered and with some uncertainty sources. Thus, the state of detected object should be estimated using an optimal filter. To do so, the state distribution is assumed Gaussian. Thus the state vector is estimated by a Kalman filter at each time step. The estimation of the probabilistic distribution can be shown using an error ellipse for a constant collision probability. Analytical forms of error ellipses can be obtained by quadratic inequalities. These quadratic inequalities make the optimization problem nonconvex. Thus, these inequalities are relaxed by applying a linearization approach. Finally, the optimization problem is reformulated to a convex optimization problem. There are some strong algorithms for solving a convex optimization problem, so the consequent path planning method can be solved efficiently with considerable performance that will be obtained in the end of this paper.

Keywords: Stochastic Path Planning, Path Planning Under Uncertainty, Linear Time-Varying MPC, Quadratic Programming, Error Ellipse, Convex Optimization.

Submission date: 23, Sep., 2013

Conditionally acceptance date: 13, June, 2015

Acceptance date: 06, March, 2016

Corresponding author: Mohsen Ahmadi Mousavi

Corresponding author's address: No.4, Javid alley, Vali-E-Asr Ave, Tehran, Iran.



1. Introduction

The trajectory planning of unmanned vehicles such as UAVs¹,UGVs²and AUVs³ in a dynamic environment, has received a great interest in recent years [13, 15, 23]. In a dynamic environment the main challenge is the existence of uncertainty in the path planning strategy of an autonomous vehicle. These problems are known as stochastic path planning or path planning under uncertainty. It is necessary to consider these uncertainty sources in a real word path planning of an autonomous vehicle. Generally, path planning methods can be divided into two main categories known as myopic and nonmyopic. In the myopic methods, there isn't an implicit planning in future time intervals. Unlike the myopic methods, for nonmyopic methods, an implicit planning occurs in future time steps. Myopic methods include graph-based method knows as graph search algorithm [1], Fuzzy control [2], artificial potential field [3], evolutionary algorithms [4] and etc. Unlike the myopic methods, there is an implicit programming based the future evolution of the system in the nonmyopic methods. Generally, the nonmyopic methods have a better input control but in these methods, the complexity of the problem increase by increasing the number of the horizon steps. Model predictive control (MPC) is the most popular approach of nonmyopic method.

Model predictive control (MPC) also known as receding horizon control (RHC) is a feedback control method which is suitable for the control of multivariable systems. MPC is a nonlinear control policy that handles input and output constraints as well as various other objective functions. The capability of handling constraints in a systematic way makes MPC a very attractive control strategy. Particularly, in those applications where the process is required to work in wide operating regions and close to the boundary of admissible states and input sets, which are imposed by constraints. In MPC, a model of plant is used to predict the future evolution of the system [5]. Based on this prediction, an optimization problem is solved at each time step to determine a plan of action over a fixed time horizon. The first input from this plan is applied to the system, such that at the next time step, a new optimization problem is solved with the time horizon shifted one step forward. The graphical representation of this procedure is obtained in the Fig. 1 for N time-steps horizon.

MPC can be used for several types of control and estimation problems including tracking problems, regulator problems and stochastic control problems. It has a variety of applications such as industrial and chemical process control [6], supply chain management [7], stochastic control in economic and finance [8], revenue management [9], control of hybrid vehicles [10], automotive applications [11] and aerospace applications [12].

In the context of autonomous control systems, the vehicle must perform target tracking as well as obstacle avoidance. For these types of problems, several solutions have been proposed in the literature such as potential field [13], A* with visibility graph [14], nonlinear trajectory generation [15], vertex-graph algorithm [16] and mixed integer linear programming [17]. Since MPC can systematically handle some constraints such as vehicle dynamics, envelop limitations and no-fly zones, it can be a suitable technique for path planning of an autonomous system [18]. In [19], [20] a predictive controller is used for minimal risk motion planning in the presence of both dynamic and static obstacles. Falcone and colleagues [11] presented two formulations for MPC to control an active front steering system in an autonomous vehicle. The first, uses a nonlinear vehicle model to predict the future evolution of the system, thus the resulting MPC requires a nonlinear optimization problem. The second formulation is a successive linearization of the nonlinear vehicle model at each time step resulting in a linear time varying MPC. It is a suboptimal MPC of the original nonlinear problem. An autonomous exploration algorithm that is suitable for urban navigation, but it is not limited to urban navigation is proposed in [15]. This algorithm is a combination of MPC-based obstacle avoidance and local obstacle map which is built using onboard laser scanners. In [21], a MPC is used for controlling a multivehicle system to track multiple target points.

Since some uncertainty sources exist in a dynamic environment, the perception of target location has some uncertainty. Thus this setting gives rise to a complex stochastic optimal control problem. Although it can be solved by dynamic programming it is still a computationally intractable approach. Bemporad and colleagues in [22] introduced a decentralized linear time varying MPC⁴ to control a fleet of UAVs. Each vehicle is a quadrotor type which is stabilized by the controller at the lower level around desired set points. These set points are generated from the high level LTV-MPC trajectory planner with a slower rate.

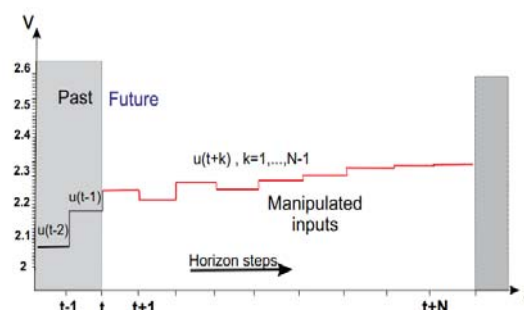


Fig. 1. Graphical representation of MPC planning method for N time-steps horizon

In the more complex situation, the autonomous vehicle must perform target tracking as well as obstacle avoidance. In other words, targets and obstacles can move stochastically in the environment. The existence of obstacles makes the feasible region nonconvex inherently. Some optimization methods which were used in the previous works for obstacle avoidance are nonconvex, and therefore most of them are greedy problem, that is, there is not a planning strategy to move because the inherent nonconvex feasible region makes the problem nonconvex entirely which would be complicated to be solved as a real time optimization problem.

Furthermore, most of the previous works are related to deterministic and semi-deterministic environments just as laboratory environment and so on. However, the main mission of an autonomous vehicle is moving in an unknown and dynamic environment where does not exists any information about the incoming future events. Therefore, it is obvious that the greedy problems fail in these situations, so it is necessary to involve the prediction of future environment states into the process of determining current input control. Indeed, model predictive control uses a systematic structure to create a sequence of inputs based on the prediction of future states of vehicle and environmental objects.

In this paper we introduce a path planning method for stochastic target tracking and obstacle avoidance in a dynamic environment. In fact, the main contribution of this paper is making the feasible region linear in order to the entire MPC problem to be converted to LTV-MPC problem firstly and secondly it can be converted to a close-form MPC problem based on the new state vector and new input control vector over the horizon. Finally, the last MPC problem is converted to close form optimization problem as authors have mentioned in [26].

The reminding of this paper has the following order. First, in the next section, the modeling of the problem is described followed by the LTV-MPC modeling for target tracking and obstacle avoidance mission. In the next section i.e. section 4, constraints of obstacle avoidance mission are described, also a desired linearization approach is proposed in details. Finally, the consequent model of predictive control is converted to a convex optimization problem in the final section of this paper.

2. Problem Modeling

Two coordinate frames are defined for navigation of an autonomous vehicle, generally. The first one is fixed to the vehicle called the vehicle-fixed reference frame. The origin of the vehicle-fixed frame is chosen to coincide with the center of gravity, which is in the principal plane of symmetry. The position and

orientation of the vehicle are described relative to this inertial reference frame. In this section, a dynamic model of the vehicle, target and obstacle are presented. Then, a measurement model is used for estimation of the object's state by Kalman filter [24]. Based on this state estimation, a trajectory prediction method is used for predicting the future evolution of the vehicle, target, and moving obstacles.

2.1. Vehicle Dynamic Modeling

In this paper, a point-mass model is used for the motion modeling of the vehicle, target and obstacles. The size of each ones is modeled by a circle surrounding the center point of gravity. Therefore, it is assumed that the size of the obstacle is known or can be computed by the on-board sensor array of the vehicle.

Since most earth-bound vehicles have dynamics that naturally decouple into vertical and horizontal planes, vehicle dynamics will be restricted to a plane. These kinds of systems can be modeled with "unicycle" dynamics by direction and speed controls. The unicycle dynamics are typically written in Cartesian coordinates as follows [23]:

$$\begin{aligned}\dot{\mathbf{x}}_{\text{vel}} &= v \cos \theta \\ \dot{\mathbf{y}}_{\text{vel}} &= v \sin \theta \\ \omega &= \dot{\theta}\end{aligned}\quad (1)$$

Where $\mathbf{x}_{\text{vel}} = [x_{\text{vel}}, y_{\text{vel}}]^T$ is the point-mass location of the vehicle and $\mathbf{u} = [v, \theta]$ is the input vector consists of velocity and heading. The discrete-time state-space of unicycle dynamic model has the following form:

$$\mathbf{x}_{\text{vel}}(k+1) = \mathbf{A}_k \mathbf{x}_{\text{vel}}(k) + \mathbf{B}_k \mathbf{u}_k(k) \quad (2)$$

where \mathbf{A}_k and \mathbf{B}_k are the transition and input matrix respectively, which are obtained as follows:

$$\begin{aligned}\mathbf{A}_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \mathbf{B}_k &= \begin{bmatrix} T_s \cos(\theta(k)) & -T_s v \sin(\theta(k)) \\ T_s \sin(\theta(k)) & T_s v \cos(\theta(k)) \end{bmatrix}\end{aligned}\quad (3)$$

2.2. Vehicle Trajectory Prediction

The state of the vehicle according to environmental disturbance such as wind has some uncertainties; these uncertainties can be modeled by a Gaussian distribution. Thus, the discrete-time and linear dynamic of vehicle in an unknown environment can be modeled as follows:

$$\mathbf{x}_{\text{vel}}(k) = \mathbf{A}_k \mathbf{x}_{\text{vel}}(k-1) + \mathbf{B}_k \mathbf{u}(k) + \nu_k \quad (4)$$

therefore, the state of the vehicle can be estimated by an extended Kalman filter (EKF) by giving the previous estimated state and other measurements from IMU⁵, GPS⁶ and on-board sensors. In other words, it

estimates the new state of vehicle from given measurements at $(k + 1)$ th time step. Suppose the function f represents the nonlinear dynamic model of the vehicle i.e.

$$\mathbf{x}_{\text{vel}} = f(\mathbf{u}) \quad (5)$$

thus, the vehicle trajectory prediction based on EKF state estimation over future horizon N can be computed as follows:

$$\mathbf{x}_{\text{vel}}^{\text{pred}} = \begin{bmatrix} \mathbf{x}_{\text{vel}}(k+1|k) \\ f(\mathbf{x}_{\text{vel}}(k+1|k), \mathbf{u}(k+1)) \\ f(f(\mathbf{x}_{\text{vel}}(k+1|k), \mathbf{u}(k+1)), \mathbf{u}(k+2)) \\ \vdots \\ f(\dots f(\mathbf{x}_{\text{vel}}(k+1|k) \dots \mathbf{u}(k+N-1)) \end{bmatrix} \quad (6)$$

2.3. Object Dynamic Modeling

In this paper, the dynamic model of the moving object (target or obstacle) is assumed to have a constant acceleration motion [25], and it has the following discrete-time state-space model:

$$\mathbf{x}_{\text{obj}}(k+1) = \mathbf{F} \mathbf{x}_{\text{obj}}(k) + \eta_k \quad (7)$$

where \mathbf{x}_{obj} is the object state vector. The transition state matrix \mathbf{F} is obtained as follows:

$$\mathbf{F}_x = \mathbf{F}_y = \begin{bmatrix} T_s/2 & T_s & 1 \\ 0 & T_s & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_x & 0 \\ 0 & \mathbf{F}_y \end{bmatrix} \quad (8)$$

In the equation (7) η_k is white Gaussian noise that models the uncertainty on the process with the following covariance matrix:

$$\mathbf{Q}_x = \mathbf{Q}_y = q \begin{bmatrix} T_s^5/20 & T_s^4/8 & T_s^3/6 \\ T_s^4/8 & T_s^3/3 & T_s^2/2 \\ T_s^3/3 & T_s^2/2 & T_s \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x & 0 \\ 0 & \mathbf{Q}_y \end{bmatrix} \quad (9)$$

where q is the power spectral density.

2.4. Measurement Model

Objects in the environment, including targets and obstacles are detected using the vehicle's on-board sensors. The location and angle of each sensor is known: $\Gamma_i = [X_i(t), Y_i(t), \theta_i(t)]$. Each sensor site named i , is assumed to make distance and direction observations to the target or obstacle as:

$$\begin{bmatrix} z_i^r(k) \\ z_i^\theta(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{ig} - X_i)^2 + (y_{ig} - Y_i)^2} \\ \arctan\left(\frac{x_{ig} - X_i}{y_{ig} - Y_i}\right) - \theta(k) \end{bmatrix} + \begin{bmatrix} \omega_i^r(k) \\ \omega_i^\theta(k) \end{bmatrix} \quad (10)$$

where the random vector $\omega_i(k) = [r_i^r, r_i^\theta]$ describes the noise in the observation process due to both error and uncertainty in the observation as the two main uncertainty sources in dynamic environments. Observation noise is taken to have a zero mean and variance as below:

$$\mathbf{R}_{r\theta}(k) = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (11)$$

It should be noted that observation is strongly range dependence and is lined up with the sensor bore-sight. Since observations are related to the vehicle-fixed frame, the observation model is nonlinear and state estimation must be executed using an extended Kalman filter. Thus information can be converted to the inertial reference frame and used for object state estimation by a linear Kalman filter. The relations below describe this mapping from the vehicle-fixed frame coordinate to the inertial reference frame.

$$\begin{bmatrix} z_i^x(k) \\ z_i^y(k) \end{bmatrix} = \begin{bmatrix} X_i(k) + z_i^r(k) \cos(z_i^\theta(k)) \\ Y_i(k) + z_i^r(k) \sin(z_i^\theta(k)) \end{bmatrix} \quad (12)$$

$$\mathbf{R}^{xy}(k) = \mathbf{Rot}(z_i^\theta(k)) \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & (z_i^\theta(k))^2 \sigma_\theta^2 \end{bmatrix} \mathbf{Rot}^T(z_i^\theta(k))$$

where $\mathbf{Rot}(\theta)$ is the rotation matrix:

$$\mathbf{Rot}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (13)$$

2.5. Object Trajectory Prediction

Using the global measurement vector computed in the previous section, the object state can be estimated by a linear Kalman filter (KF). Thus, the future evolution of the object state (target or obstacle) can be predicted over the N time-steps horizon. For this purpose, an iterative execution of state estimation step leads to the object trajectory prediction which it is represented by the state estimation vector $\mathbf{x}_{\text{obj}}^{\text{pred}}$ that can be computed as follows:

$$\mathbf{x}_{\text{obj}}^{\text{pred}} = \begin{bmatrix} x_{\text{obj}}(k+1|k) \\ \mathbf{F}x_{\text{obj}}(k+1|k) \\ \vdots \\ \mathbf{F} \dots \mathbf{F}x_{\text{obj}}(k+1|k) \end{bmatrix} \quad (14)$$

3. LTV-MPC Modeling

General concepts of LTV-MPC and its stability analysis are illustrated in [24]. Based on these materials and definitions, the LTV-MPC formulation that is suitable for the main goal of this paper, i.e. target tracking and obstacle avoidance in a dynamic environment, can be modeled as follows:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1} \left\| \mathbf{Q}_k \mathbf{x}_{\text{vel}}(t+k|t) - \mathbf{x}_{\text{tg}}(t+k|t) \right\|^2 + \left\| \mathbf{R}_k \Delta \mathbf{u}(t+k|t) \right\|^2 \\ \text{s.t.} \quad & \mathbf{x}_{\text{vel}}(k+1) = \mathbf{A}_k \mathbf{x}_{\text{vel}}(k) + \mathbf{B}_k \mathbf{u}(k), \quad k=0, \dots, N-1 \\ & \Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u}(t+k|t) \leq \Delta \mathbf{u}_{\max}, \quad k=1, \dots, N \\ & \mathbf{u}_{\min} \leq \mathbf{u}(t+k|t) \leq \mathbf{u}_{\max}, \quad k=0, \dots, N-1 \end{aligned} \quad (15)$$

where the first term of the objective function is target tracking criteria that is equivalent to $(\mathbf{x}_{\text{vel}}(t+k|t) - \mathbf{x}_{\text{tg}}(t+k|t))^T \mathbf{Q}_k (\mathbf{x}_{\text{vel}}(t+k|t) - \mathbf{x}_{\text{tg}}(t+k|t))$ and \mathbf{Q}_k is its weight, $\mathbf{x}_{\text{vel}}(t+k|t)$ is the predicted position of target and it is computed by the target

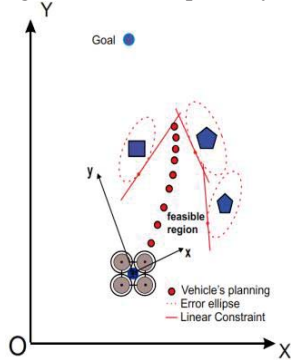


Fig. 2. linearization of error ellipses related to detected obstacles

trajectory prediction procedure described in the section 2.5. The second term is the set-point weighting criteria with the weight of \mathbf{R}_k and it is equivalent to $\Delta \mathbf{u}(t+k|t)^T \mathbf{R}_k \Delta \mathbf{u}(t+k|t)$. $\Delta \mathbf{u}(t+k|t)$ is the predicted set-point increment. The first constraint is the linearization dynamic modeling of the vehicle and the next two are predicted input increment constraints, and predicted set-point constraints, that must be adhered to the vehicle maneuverability constraints, respectively.

4. Linear Constraints for Obstacle Avoidance

In the presence of obstacles, the feasible space for the movement of autonomous vehicles becomes non-convex. Thus, the equivalent optimization-based path planning method is non-convex and hence difficult to solve. Additionally, the existence of some uncertainty sources in the dynamic environment causes probabilistic obstacle avoidance constraints in the optimization-based path planning problem. Therefore, the computational complexity increases. The main goal

of this paper is the conversion of these nonlinear probabilistic constraints to linear deterministic constraints. It is shown that using this procedure, the final MPC consists of a quadratic objective function and multiple linear constraints. Thus the resulting optimization problem is convex and easy to solve by means of some available algorithms.

In the previous section the state of the obstacle was assumed to be a Gaussian random variable. It was estimated and then predicted using the optimal Gaussian estimation filter i.e., kalman filter.

The geometric representation of this random variable for a constant probability is a point with an ellipse around it, known as the error ellipse. This point that is the center of the ellipse is the mean vector of the Gaussian random variable and the covariance matrix of the Gaussian random variable consisting of information about the semi axis of the error ellipse. In other words, the error ellipse is the probability distribution of the obstacle's state for a constant probability δ . Based on the concept of error ellipse [27], the analytical form of this ellipse is given as below:

$$\{(x, y) : \mathbf{x}^T \mathbf{T}_{\text{obs}} \mathbf{D}_{\text{obs}}^{-1} \mathbf{T}_{\text{obs}}^T \mathbf{x} - \mathbf{k}_{\text{obs}} = 0\} \quad (16)$$

where \mathbf{T}_{obs} , \mathbf{D} and \mathbf{k} are computed as follows:

$$\begin{aligned} \mathbf{T}_{\text{obs}} &= [\mathbf{v}_1 | \mathbf{v}_2] \\ \mathbf{D}_{\text{obs}} &= \text{diag}(\lambda_1, \lambda_2) \end{aligned} \quad (17)$$

$$\mathbf{k}_{\text{obs}} = -2 \ln(1 - \delta)$$

\mathbf{v}_1 and \mathbf{v}_2 are the eigenvectors corresponding to the eigenvalues λ_1 and λ_2 . The semi-axis of the ellipse can be computed as follows:

$$\begin{aligned} a &= \sqrt{\mathbf{k} \times \lambda_1} \\ b &= \sqrt{\mathbf{k} \times \lambda_2} \end{aligned} \quad (18)$$

Thus probabilistic constraints are transmitted to the deterministic and nonconvex ones as follows:

$$\mathbf{x}_{\text{vel}}^T \mathbf{T}_{\text{obs}} \mathbf{D}_{\text{obs}}^{-1} \mathbf{T}_{\text{obs}}^T \mathbf{x}_{\text{vel}} - \mathbf{k}_{\text{obs}} \geq 0 \quad (19)$$

The size of the vehicle and the obstacle are considered by a circle that surrounds each one. For this purpose, it is assumed that the size of the obstacles can be estimated using the on-board sensors of the vehicle. If r_{vel} and r_{obs} are the radius of the vehicle and the obstacle, respectively, the error ellipse for the point mass control of the vehicle expands, and the new semi-axis lengths of it are computed as follows:

$$\begin{aligned} a_{\text{exp}} &= a + r_{\text{vel}} + r_{\text{obs}} \\ b_{\text{exp}} &= b + r_{\text{vel}} + r_{\text{obs}} \end{aligned} \quad (20)$$

It means the size of both vehicle and obstacle are transmitted to the semi-axis of the error ellipse. In addition, this definition of size can solve the problem of sharp shapes in path planning problems. That means



an obstacle with a sharp corner or shape is simplified to a circle and consequently a smooth path for the vehicle can be generated. Although using the above procedure, probabilistic constraint is converted to a deterministic one; the non-convexity of the search space is still a problem yet. The main idea is the linearization of the search space at each time step. To reach this goal, in the first step, a line passing from the gravity center of vehicle to gravity center of obstacle, intersected with the expanded error ellipse is drawn. In the second step, a tangent line at this intersection point is computed and used as the boundary of the optimization problem

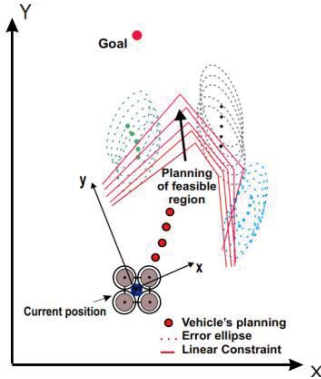


Fig. 3. planning of linearized feasible region

constraint. It should be noted that at each time step, the outer region of the error ellipse and this line is the valid space for the vehicle to move in. Thus at each time step, linear constraints exist for avoiding an obstacle as follows:

$$\mathbf{A}_o(t)\mathbf{x}_{vel}(t) \leq b_o(t) \quad (21)$$

where $\mathbf{A}_o(t)$ is a row vector and $b_o(t)$ is a scalar one. In the Fig. 3 the linearization of expanded error ellipses for multiple obstacles with different size and shape are obtained. The intersection of valid regions for avoiding each obstacle makes a feasible region for movement of the vehicle and can be represented as follows:

$$\mathbf{A}_o^i(t)\mathbf{x}_{vel}(t) \leq b_o^i(t) \quad i = 1, 2, \dots, l \quad (22)$$

where l is the number of detected obstacles at each time step. As described, a prediction of the moving obstacle is executed at each time step. There are N linear constraints based on the number of horizon steps. Furthermore, for l detected obstacles and N horizon steps, there are $l*N$ linear constraints in the optimization modeling of the MPC problem as follows:

$$\mathbf{A}_o^i(k)\mathbf{x}_{vel}(k+t) \leq b_o^i(k) \quad i = 1, 2, \dots, l, \quad k = 1, 2, \dots, N \quad (23)$$

Moreover, as it is standard in all practical MPC, the slack variable ε is used to soften the above obstacle avoidance constraints. Upon adding the term $\rho\varepsilon^2$ to the objective function, the violation of the constraints on the position of the vehicle is penalized. Thus, the final LTV-MPC that is used has the following form:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1} \left\| \mathbf{Q}_k \mathbf{x}_{vel}(t+k|t) - \mathbf{x}_{ig}(t+k|t) \right\|^2 + \left\| \mathbf{R}_k \Delta \mathbf{u}(t+k|t) \right\|^2 \\ & + \rho \varepsilon^2 \\ \text{s.t.} \quad & \mathbf{x}_{vel}(k+1) = \mathbf{A}_k \mathbf{x}_{vel}(k) + \mathbf{B}_k \mathbf{u}(k), \quad k = 0, \dots, N-1 \\ & \Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u}(t+k|t) \leq \Delta \mathbf{u}_{\max}, \quad k = 1, \dots, N \\ & \mathbf{u}_{\min} \leq \mathbf{u}(t+k|t) \leq \mathbf{u}_{\max}, \quad k = 0, \dots, N-1 \\ & \mathbf{A}_o(k)\mathbf{x}_{vel}(k+t) \leq b_o(k) + \varepsilon, \quad k = 1, \dots, N \end{aligned} \quad (24)$$

where ρ is the weight coefficient of penalizing.

5. Convex Optimization Modeling of LTV-MPC

A Convex optimization problem is an easy problem in the comparison of a non-convex problem in the mathematic optimization problems. Moreover, in [26] a very fast algorithm has been purposed for a convex optimization problem with quadratic objective function and linear constraints. As mentioned in this article, the MPC problem can be solved in the millisecond scale. Solving the tracking problem in the kilohertz causes the stability of the vehicle and also decreasing the uncertainty of the vehicle about its environment. Since the decision making time can be chosen in the millisecond scale. For Solving the MPC problem in equation (24) using convex optimization, it must be reformulated as a close form of the input vector and increment input vector. For this purpose, the input vector and increment input vector are defined as $\mathbf{U}_s(t) = [\mathbf{u}(t), \mathbf{u}(t+1), \dots, \mathbf{u}(t+N-1), \varepsilon]^T$ and $\Delta \mathbf{U}_s(t) = [\Delta \mathbf{u}(t), \Delta \mathbf{u}(t+1), \dots, \Delta \mathbf{u}(t+N-1), \varepsilon]^T$ respectively. Moreover $\mathbf{U}(t) = [\mathbf{u}(t), \mathbf{u}(t+1), \dots, \mathbf{u}(t+N-1)]^T$ introduces the new input vector and $\Delta \mathbf{U}(t) = [\Delta \mathbf{u}(t), \Delta \mathbf{u}(t+1), \dots, \Delta \mathbf{u}(t+N-1)]^T$ introduces the new increment input vector. The relation between new input vector and old one is expressed by the matrix Γ and the relation between the slack variable and old input vector is expressed by the matrix Γ_c , respectively as follows:

$$\mathbf{U}(t) = \Gamma \mathbf{U}_s(t) \quad (25)$$

$$\varepsilon = \Gamma_c \mathbf{U}_s(t)$$

where coefficient matrices are defined as:

$$\begin{aligned} \Gamma &= [\mathbf{I}_{N \times N} \mid \mathbf{0}_{N \times 1}] \\ \Gamma_c &= [\mathbf{0}_{N \times N} \mid \mathbf{1}_{N \times 1}] \end{aligned} \quad (26)$$

\mathbf{x}_{ig} is the trajectory prediction vector of the target and \mathbf{x}_{vel} is the trajectory prediction of the vehicle that must be computed by the optimization problem. Based on this definition, the problem in equation (24) can be modeled as follows:

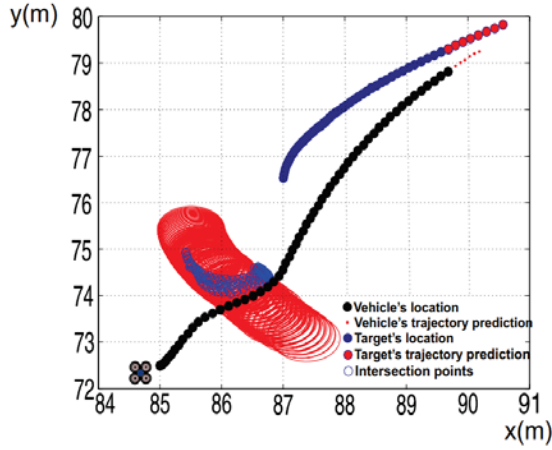


Fig. 4. LTV-MPC based target tracking and obstacle avoidance for the N=8 time-steps horizon

$$\begin{aligned}
 \min \quad & (\mathbf{X}_{\text{vel}} - \mathbf{X}_{\text{tg}})^T \mathbf{Q} (\mathbf{X}_{\text{vel}} - \mathbf{X}_{\text{tg}}) + \Delta \mathbf{U}_s^T \mathbf{R} \Gamma \Delta \mathbf{U}_s + \Delta \mathbf{U}_s^T (\Gamma_c)^T \rho \Gamma_c \Delta \mathbf{U}_s \\
 \text{s.t.} \quad & \mathbf{X}_{\text{vel}} = \mathbf{S} \Gamma \mathbf{U}_s + \mathbf{T} \mathbf{x}_{\text{vel}}^0 \\
 & \mathbf{U}_{\min} \leq \Gamma \mathbf{U}_s \leq \mathbf{U}_{\max} \\
 & \Delta \mathbf{U}_{\min} \leq \Gamma \Delta \mathbf{U}_s \leq \Delta \mathbf{U}_{\max} \\
 & \mathbf{A}_{\text{rep}}^i \mathbf{X}_{\text{vel}} \leq \mathbf{b}_{\text{rep}}^i + \Gamma^c \Delta \mathbf{U}_s, \quad i = 1, 2, \dots, l
 \end{aligned} \quad (27)$$

Where

$$\begin{aligned}
 \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}_1 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{Q}_N \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & 0 & 0 & 0 \\ 0 & \mathbf{R}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{R}_N \end{bmatrix} \\
 \Gamma &= [\mathbf{I}_{N \times N} \mid \mathbf{0}_{N \times m}], \quad \Gamma^c = [\mathbf{0}_{N \times (N \times m)} \mid \mathbf{I}_{N \times 1}] \\
 \mathbf{S} &= \begin{bmatrix} \mathbf{A} & 0 & 0 & 0 & 0 \\ \mathbf{AB} & \mathbf{A} & 0 & 0 & 0 \\ \mathbf{A}^2 \mathbf{B} & \mathbf{AB} & \mathbf{A} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{N-1} \mathbf{B} & \mathbf{A}^{N-2} \mathbf{B} & \dots & \mathbf{AB} & \mathbf{A} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}
 \end{aligned} \quad (28)$$

This problem can be modeled as a function of the input increment vector $\Delta \mathbf{U}_s$ by means of the variable definition $\mathbf{x}_u(t) \equiv \mathbf{u}(t-1)$. The new state vector is $\mathbf{x}_{\text{new}} \equiv [\mathbf{x}(t), \mathbf{x}_u(t)]^T$ and consequently the new dynamic model of the vehicle has the following form:

$$\begin{aligned}
 \mathbf{x}_{\text{new}}(t+1) &= \mathbf{A}_{\text{new}} \mathbf{x}_{\text{new}}(t) + \mathbf{B}_{\text{new}} \Delta \mathbf{u}(t) \\
 \mathbf{A}_{\text{new}} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_{\text{new}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{I} \end{bmatrix}
 \end{aligned} \quad (29)$$

Also the relation between the new state vector, \mathbf{x}_{new} , the state vector of the vehicle \mathbf{x}_{vel} , and the input vector \mathbf{U}_s is as follows:

$$\begin{aligned}
 \mathbf{X}_{\text{vel}} &= \mathbf{M} \mathbf{X}_{\text{new}} \\
 \Gamma \mathbf{U}_s &= \mathbf{M}^c \mathbf{X}_{\text{new}} \\
 \mathbf{M} &= \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times m} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \mathbf{I}_n & \mathbf{0}_{n \times m} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbf{I}_n & \mathbf{0}_{n \times m} \end{bmatrix} \\
 \mathbf{M}^c &= \begin{bmatrix} \mathbf{0}_{m \times n} & \mathbf{I}_m & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \mathbf{0}_{m \times n} & \mathbf{I}_m & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbf{0}_{m \times n} & \mathbf{I}_m \end{bmatrix}
 \end{aligned} \quad (30)$$

By substituting the above relation in equation (27) and simplifying, the closed form and final convex optimization problem can be formulated as follows:

$$\begin{aligned}
 \min \quad & \frac{1}{2} \Delta \mathbf{U}^T \mathbf{H} \Delta \mathbf{U} + \mathbf{F} \Delta \mathbf{U} \\
 \text{s.t.} \quad & \Delta \mathbf{U}_{\min} \leq \Gamma \Delta \mathbf{U}_s \leq \Delta \mathbf{U}_{\max} \\
 & \mathbf{U}_{\min} - \mathbf{M}^c \mathbf{T} \mathbf{x}_{\text{new}}^0 \leq \mathbf{M}^c \mathbf{S} \Gamma \Delta \mathbf{U}_s \leq \mathbf{U}_{\max} - \mathbf{M}^c \mathbf{T} \mathbf{x}_{\text{new}}^0 \\
 & \mathbf{G}_i \Delta \mathbf{U}_s \leq \mathbf{w}_i - \mathbf{P}_i \mathbf{x}_{\text{new}}^0
 \end{aligned} \quad (31)$$

where coefficient matrices are computed as follows:

$$\begin{aligned}
 \mathbf{H} &= \begin{bmatrix} \mathbf{S}^T \mathbf{M}^T \mathbf{Q} \mathbf{M} \mathbf{S} + \mathbf{R} & 0 \\ 0 & \rho \end{bmatrix} \\
 \mathbf{F} &= (2 \mathbf{x}_{\text{tg}}^T \mathbf{T}^T \mathbf{M}^T \mathbf{Q} \mathbf{M} \mathbf{S} - 2 \mathbf{x}_{\text{tg}}^T \mathbf{Q} \mathbf{M} \mathbf{S}) \Gamma \\
 \mathbf{G}_i &= \mathbf{A}_i \mathbf{M} \mathbf{S} \Gamma - \Gamma^c \\
 \mathbf{w}_i &= \mathbf{b}_i \\
 \mathbf{P}_i &= \mathbf{A}_i \mathbf{M} \mathbf{T}
 \end{aligned} \quad (32)$$

6. Simulations and Results

The simulation of the purposed method is executed by the CVX toolbox [28] in MATLAB. To start, the initial state of the target, obstacle and vehicle are selected $\mathbf{x}_{\text{tg}}^0 = [87, 1, 0, 86.5, 1, .05]^T$, $\mathbf{x}_{\text{obs}}^0 = [85.5, 0, 1, 75.5, -.8, -.9]^T$ and $\mathbf{x}_{\text{vel}}^0 = [85, 72.5]^T$ respectively. The decision making period is $T_s = .05$ and the covariance matrix of the sensor noise, that is strongly range dependent, is assumed as follows:

$$\mathbf{R} = \begin{bmatrix} 0.1 r & 0 \\ 0 & 0.05 \varphi \end{bmatrix} \quad (33)$$

Boundary vectors of the input vector are assumed as $\Delta \mathbf{u}_{\max} = [0.5, 0.15]^T$, $\Delta \mathbf{u}_{\min} = -\Delta \mathbf{u}_{\max}$, $v_{\max} = 5$ and $v_{\min} = -v_{\max}$. Finally the size of the vehicle and obstacle are determined by circles with radius 0.3 and 0.2 respectively and the collision probability δ is assumed 0.7.

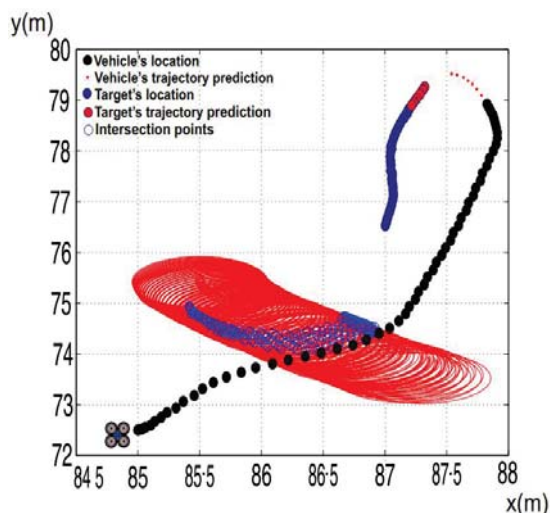


Fig. 5. LTV-MPC based target tracking and obstacle avoidance for the N=8 time-steps horizon

The simulation results are shown in the Fig. 4 and Fig. 5 for two different simulations using CVX toolbox in Matlab. Since the motion of obstacle and target are modeled stochastically, motion of target, obstacle and consequently motion of vehicle must be different for separate simulations. Based on these results, it can be understood that the proposed method is independent of the behavior of objects in dynamic environments and can be used for planning of an autonomous vehicle in cluttered environments.

As can be seen, for both figures, autonomous vehicles have suitable planning to track and catch targets in future time steps. It should be noted that the stochastic behavior and movement of vehicle in the real dynamic environment, is considered in these simulations. Also from both simulation results, the performance of linearization for obstacle avoidance mission can be investigated and satisfied, this is clear from the comparison between vehicle trajectory and intersection points set that are shown in the both simulation results in the Fig. 4 and Fig. 5. Moreover, the planning of the vehicle in horizon steps is shown at the last time step for both simulations.

The angle and velocity of second simulation are represented in Fig. 6 and Fig. 7 respectively as time goes on. As can be seen, the velocity curve is limited to the 5 m/s i.e. the upper boundary of the velocity as one of two control variables.

Also the estimated distance of the vehicle from target is obtained in the Fig. 8. As it can be seen, this curve have a local maximum which is the effect of detected obstacle in the path of vehicle and it can be interpreted as the reaction of vehicle against the presence of a stochastic and moving obstacle. Moreover, the curve of set point cost

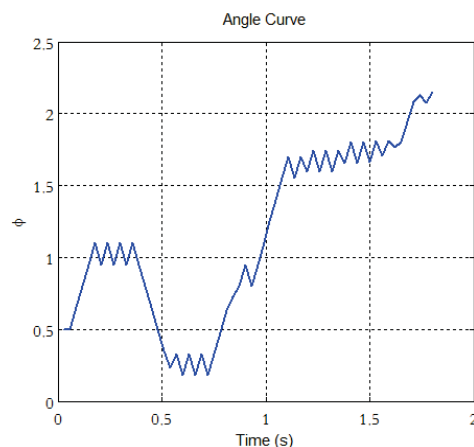


Fig. 6. The angle control variable

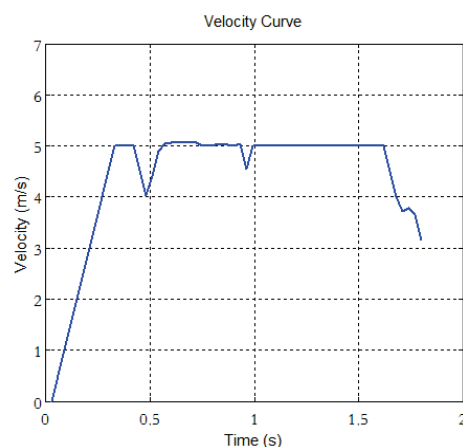


Fig. 7. The velocity control variable

is obtained in the Fig. 9. It can be interpreted as the control effort of vehicle to achieve to the desired goal i.e. target tracking and obstacle avoidance in a real and dynamic environment in presence of some uncertainty sources. In the comparison with the related work in the literature, the introduced method in this paper has two main advantages. First, the inherent stochastic problem is transformed to a deterministic one using the concept of error ellipse in the probability theory. Second, the nonconvexity of problem is relaxed using a successive linearization approach. Thus we achieve a useful deterministic and convex optimization method for path planning of the autonomous vehicle in a cluttered environment in the presence of some uncertainty sources. Finally, for the consequent convex optimization based path planning, there are some strong algorithms to solve this problem in real time. Furthermore, in this paper, we proposed an online optimization based path planning approach.

Generally, this linearization procedure has two main effects. The first one is related to decreasing of

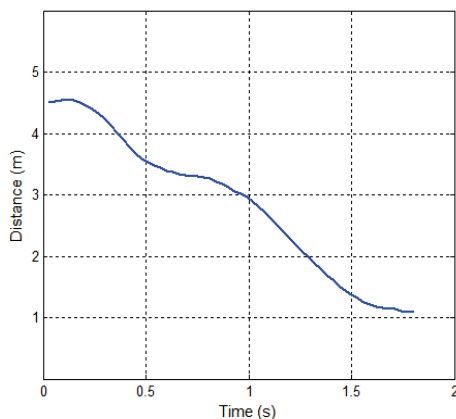


Fig. 8. Estimated Distance from the target

linearization error in the kinematic model and the second one is related to the accuracy and complexity of the problem compared to original and greedy problems. As mentioned, the linearization makes a very fast MPC problem which can be solved in millisecond scale, so the limitation of processing on the decision making period is eliminated so that the linear kinematic would be very close to the original nonlinear one over time.

There are multiple criteria to examine the effect of feasible region linearization including accuracy, complexity and flexibility in dynamic environment. It should be stressed that effect of linearization must be examined from all of these criteria jointly not separately. From the Fig. 2, it is clear that the linearization makes the feasible region limited, but it makes the original problem easy in order to make it suitable for a real time application such as path planning problem. On the other hand, this linearization makes use of planning strategy in motion control of vehicles possible compared to greedy method which considers only one horizon step which obtained in the Fig. 2. This planning strategy creates flexibility in controller for dynamic environments due to state prediction in the future time-steps which obtained in the Fig. 3. Although there is difference between the original feasible space and linearized version in the Fig. 2, the sequence of predicted linear feasible regions over horizon steps as obtained in the Fig. 3 is a good interpretation of original feasible region at each time step, that is, we have a planning of linearized feasible region over horizon steps. As can be seen, the free corner of original feasible space is followed by sequence of linear feasible space over the horizon in the planning strategy. It can be interpreted that these sequence of linear constraints are good replacement of quadratic constraints for greedy problem.

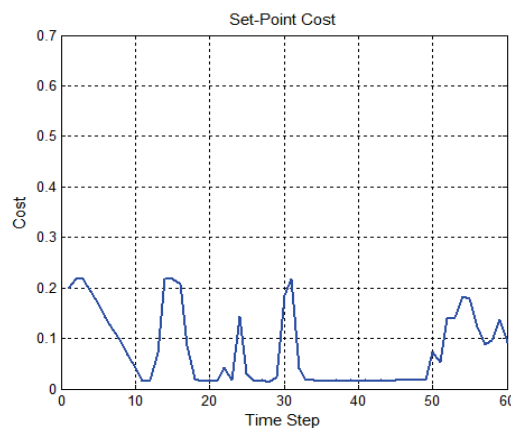


Fig. 9. Set-Point cost function

7. Conclusion and Future Work

In this paper we combine a model predictive control and the linear quadratic regulator problem to plane efficiency the trajectory motion of an autonomous vehicle in presence of uncertainty sources in a dynamic and cluttered environment. The main contribution of this paper is the convex optimization modeling of the proposed planning method. To do so, first the probabilistic optimization problem is converted to the deterministic one and then the nonconvex constraints are relaxed to the linear ones. Finally, the consequent problem is reformulated to a convex optimization problem. Our future work is related to apply this idea to a network of vehicles using decentralized data fusion approaches. In the networked control system, the main challenge is the stability of network which must be modeled in the final convex optimization problem.

References

- [1] A. Dogan and U. Zengin, " Unmanned aerial vehicle dynamic-target pursuit by using probabilistic threat exposure map " *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 944–954, 2006.
- [2] B. M. Sathiyaraj, L. C. Jain and A. Finn, " Multiple UAVs path planning algorithms: a comparative study " *Fuzzy Optimization and Decision Making*, vol. 7, no. 3, pp. 257–267, 2008.
- [3] S. Li and X. Sun, " A real-time UAV route planning algorithm based on fuzzy logic techniques " *The Sixth World Congress on Intelligent Control and Automation*, WCICA, vol. 2, pp. 8750–8753, 2006.
- [4] S. Li and X. Sun, "FPGA implementation of genetic algorithm for UAV real-time path planning" *Unmanned Aircraft Systems*, WCICA, vol. 2, pp. 8750–8753, 2006.
- [5] S. S. Razavi, S. M. Esmailzadeh. "Boiler-Turbine System Predictive Controller Design", *Journal of Iranian Association of Electrical and Electronics Engineers*, vol. 13, no. 2, pp. 141-152, 2016.
- [6] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control*

Engineering Practice, vol. 11, no. 7, pp. 733–764, 2003.

- [7] E. G. Cho, K. A. Thoney, T. J. Hodgson, and R. E. King, "Supply chain planning: Rolling horizon scheduling of multi-factory supply chains," in *Proc. Conference Winter Simulation: Driving Innovation.*, New Orleans, LA, pp. 1409–1416, 2003.
- [8] F. Herzog, Strategic portfolio management for long-term investments. An optimal control approach by Florian Herzog. IMRT Press, Measurement and Control Laboratory, ETH, Zurich ML, 2005
- [9] K. T. Talluri and G. J. Van Ryzin. "The Theory and Practice of Revenue Management". New York: Springer-Verlag, 2004.
- [10] M. T. Nodeh, H. G. Narm, and A. Hajizadeh, "Predictive control strategy for power management in parallel hybrid-electric vehicle," *Journal of Iranian Association of Electrical and Electronics Engineers.*, vol. 13, no.3, pp. 49–56, 2016.
- [11] P. Falcone, F. Borrelli, J. Asgari, E. H. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transaction on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [12] MA. Mousavi, Z. Heshmati, and B. Moshiri, "LTV-MPC Based Path Planning of an Autonomous Vehicle Via Convex Optimization," in *IEEE proceeding of 21st Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–7, 2013.
- [13] G. Hoffmann, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," In *IEEE Conference on Decision and Control*. 2008
- [14] T. Paul, T. Krogstad, and J. Gravdahl, "Modeling of UAV formation flight using 3D potential field," *Simulation Modeling Practice and Theory*, vol. 16, no. 9, pp. 1453–1462, 2008.
- [15] W. Dunbar and R. Murray, "Model predictive control of coordinated multi-vehicle formations," in *IEEE Conference on Decision and Control*, vol. 4, pp. 4631–4636, 2002.
- [16] T. Lozano-Pérez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.
- [28] Systems and Robotics, Instituto Superior Técnico. [Online]. Available: http://hans.fugal.net/comps/papers/ribeiro_2004.pdf, 2004.
- [29] M. Grant and S. Boyd "CVX: matlab software for disciplined convex programming. Version 2.0 Beta", September 2012. [Online]. Available: <http://cvxr.com/cvx>
- [17] F. Borrelli, T. Keviczky, G. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds. Springer-Verlag, pp. 168–183, 2005.
- [18] D. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proc IEEE Conference on Decision and Control*, pp. 3621–3626, 2003.
- [19] B. Xu, D. J. Stilwell and A. J. Kurdila, "A Receding Horizon Controller for Motion Planning in the Presence of Moving Obstacles," in *2010 IEEE International Conference on Robotics and Automation.*, pp. 315–320.
- [20] MA. Mousavi, B. Moshiri, Z. Heshmati. "A New Predictive Motion Control of a Planar Vehicle under Uncertainty via Convex Optimization," *International Journal of Control, Automation and Systems*, Springer, vol. 8, no. 48, pp.1-9, 2016.
- [21] MA. Mousavi, B. Moshiri, Z. Heshmat. "Cooperative control of networked autonomous vehicles using convex optimization." in *3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pp. 681–687, 2015.
- [22] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," *IEEE Conference on Decision and Control and European Control Conference*, pp. 7488–7493, 2011.
- [23] Zh. Shaolei, Zh. Wenguang, Zh. Chao and L. Ming, "Collision-free UAV formation flight control based on nonlinear MPC". In *International Conference on Electronics, Communication and Control*, pp. 1951–1956, 2001.
- [24] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics. The MIT Press, 2005.
- [25] X. R Li and V. P. Jilkov, "Survey of maneuvering target tracking: dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, October. 2003.
- [26] Y. Wang and S. Boyd, "Fast model predictive control using online optimization", *IEEE Transaction on Control Systems Technology*, 2010.
- [27] M. I. Ribeiro. "Gaussian probability density functions: properties and error characterization", Institute for

¹ Unmanned Aerial Vehicle

² Unmanned Ground Vehicle

³ Autonomous Underwater Vehicle

⁴ LTV-MPC

⁵ Inertial Measurement Unit

⁶ Global Positioning System