LTV-MPC Based Path Planning of an Autonomous Vehicle Via Convex Optimization

Mohsen Ahmadi Mousavi*, Zainabolhoda Heshmati**, and Behzad Moshiri ***
* Faculty of New Sciences and Technologies, University of Tehran, s.m.a.mousavi@ut.ac.ir
** Faculty of New Sciences and Technologies, University of Tehran, zheshmati@ut.ac.ir
*** Faculty of Electrical and Computer Engineering, University of Tehran, moshiri@ut.ac.ir

Abstract: This paper investigates the stochastic path planning of a vehicle in the presence of some moving obstacles and uncertainty sources. Upon a successive linearization, a Linear-Time Varying Model Predictive Control (LTV-MPC) is used as the planning method. The uncertainty is modelled by a Gaussian distribution and the obstacle avoidance mission is modelled in the form of linear constraints of the LTV-MPC. Finally by applying some algebraic simplification the LTV-MPC is converted to a convex optimization problem. There are strong algorithms for solving a convex optimization problem, thus the consequent path planning method can be solved efficiently and has high performance operation that will be obtained.

Keywords: Stochastic path planning, LTV-MPC, Convex Optimization.

1. Introduction

Model Predictive Control (MPC) also known as Receding Horizon Control (RHC) is a feedback control method that 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 [1]. 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 [2], supply chain management [3], stochastic control in economic and finance [4], revenue management [5], control of hybrid vehicles [6], automotive applications [7] and aerospace applications [8].

The trajectory planning of unmanned vehicles such as UAVs¹, UGVs² and AUVs³ in a dynamic environment, has received a great interest in recent years [9,11,18]. In the context of autonomous systems control, 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 [10], A* with visibility graph [11], nonlinear trajectory generation [12], vertex-graph algorithm [13] and mixed integer linear programming [14]. 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 [15]. In [9] a model predictive controller is used for minimal risk motion planning in the presence of both dynamic and static obstacles. Using a simple partial differential equation for the time optimal control problem proposed. Falcone and colleagues [7] present two formulations of MPC for controlling 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 that is a suboptimal MPC of the original nonlinear problem. An autonomous exploration algorithm that is suitable for, but not limited to urban navigation is proposed in [16]. This algorithm is a combination of MPC-based obstacle avoidance with local obstacle map, built using onboard laser scanners. In [17] an MPC is used for controlling a multivehicle system to track multiple target points. Since some uncertainty sources

¹ Unmanned Aerial Vehicle
² Unmanned Ground Vehicle
³ Autonomous Underwater Vehicle
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 that although can be solved by dynamic programming, yet it is a computationally intractable approach. Bemporad and colleagues [18] introduced a decentralized linear time varying MPC to control a fleet of UAVs. Each vehicle is a quadrotor type which is stabilized by the controller at the lower level around the desired set points, which are generated from the high level LTV-MPC trajectory planner with a slower rate.

In the professional path planning methods the autonomous vehicle must perform target tracking as well as obstacle avoidance in the dynamic environment, meaning targets and obstacles can move stochastically in the environment. In this paper we introduce a path planning method for stochastic target tracking and obstacle avoidance in the dynamic environment. We use a combination of LTV-MPC with linear quadratic regulation (LQR) to formulate the path planning mission and solve the consequent quadratic programming using convex optimization.

In the next section, the modeling of the problem is described following by the LTV-MPC modeling for the mission of this paper. In section 4, the linear constraint modeling for the obstacle avoidance mission is described with details. Finally the consequent model of predictive control is converted to a convex optimization problem.

2. Problem Modelling

In general for navigation of an autonomous vehicle two coordinate frames are defined. The first one is fixed to the vehicle and is 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 [20]. 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 Modelling

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

As most earth-bound vehicles have dynamics that naturally decouple into vertical and horizontal planes, vehicle dynamics will be restricted to a plane. Systems of this sort can be modelled with "unicycle" dynamics by direction and speed controls. The unicycle dynamics are typically written in Cartesian coordinates as follows [21]:

\[
\begin{align*}
\dot{x}_{vel} &= v \cos \theta \\
\dot{y}_{vel} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]

(1)

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

\[
x_{vel}(k+1) = A_k x_{vel}(k) + B_k \mathbf{u}_k(k)
\]

(2)

Where \( A_k \) and \( B_k \) are the transition and input matrix respectively, which are obtained as follows:

\[
A_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

B_k = \[
\begin{bmatrix}
T_s \cos(\theta(k)) & -T_s v \sin(\theta(k)) \\
T_s \sin(\theta(k)) & T_s v \cos(\theta(k))
\end{bmatrix}
\]

(3)

Where \( T_s \) is the sampling period of the decision making.

2.2 Vehicle Trajectory Prediction

The state of the vehicle can be estimated by an Extended Kalman Filter (EKF), given the previous estimated state and other measurements from IMU\(^5\), GPS\(^6\) and other on-board sensors, i.e. estimates from given measurements at time step \( k + 1 \). If the function \( f \) represents the nonlinear dynamic model of the vehicle i.e.,

\[
x_{vel} = f(\mathbf{u})
\]

(4)

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

\(^4\text{LTV-MPC}\)

\(^5\text{Inertial Measurement Unit}\)

\(^6\text{Global Positioning System}\)
is white Gaussian noise that models the uncertainty on the process with the following covariance matrix:

\[
Q = \begin{bmatrix}
\frac{T_x^2}{20} & \frac{T_x T_y}{8} & \frac{T_y^2}{6} \\
\frac{T_x T_y}{8} & \frac{T_y^2}{3} & \frac{T_y^2}{2} \\
\frac{T_x^2}{3} & \frac{T_x T_y}{2} & T_x
\end{bmatrix}
\]

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: \( \Gamma_i = [X(i), Y(i), \theta(i)] \). Each sensor site \( i \) is assumed to make distance and direction observations to the target or obstacle as:

\[
\begin{bmatrix}
z_x' (k) \\
z_y' (k)
\end{bmatrix} = \begin{bmatrix}
\sqrt{(x_x - X)\hat{r} + (y_y - Y)\hat{r}} \\
\arctan\left(\frac{x_x - X}{y_y - Y}\right) - \theta(k)
\end{bmatrix} + \begin{bmatrix}
\omega_x'(k) \\
\omega_y'(k)
\end{bmatrix}
\]

where the random vector \( \omega = [\omega_x', \omega_y'] \) 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:

\[
R_{\omega}(k) = \begin{bmatrix}
\sigma_x^2 & 0 \\
0 & \sigma_y^2
\end{bmatrix}
\]

It should be noted that observation is strongly range dependent and is lined up with the sensor bore-sight. Since observations are relative to the vehicle-fixed frame; this information must be converted to the inertial reference frame and used for object state estimation by a linear filter. The relations below describe this mapping from the vehicle-fixed frame coordinate to the inertial reference frame.

\[
R_{\theta}(k) = \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\]

### 2.5 Object Trajectory Prediction

Using the global measurement vector computed in the previous section, the object state can be estimated by a kalman filter (KF). Thus the future evolution of the object state (target or obstacle) can be predicted over the N time-steps horizon. With an iterative application of the state estimation step using the kalman filter; it leads to the object trajectory prediction represented by the state estimation vector.

### 3. LTV-MPC Modelling

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 modelled as follows:

\[
\min \sum_{k=0}^{N-1} \left[ Q_{\text{vel}}(t+k|t) - x_{\text{vel}}(t+k|t) \right]^2 + R_{\text{vel}}(t+k|t) + \rho \varepsilon^2
\]

s.t \( x_{\text{vel}}(k+1) = A_k x_{\text{vel}}(k) + B_k u(k), \quad k = 0, \ldots, N-1 \)

\( \Delta u_{\min} \leq u(t+k|t) \leq \Delta u_{\max} \), \( k = 1, \ldots, N \)

\( u_{\min} \leq u(t+k|t) \leq u_{\max} \), \( k = 0, \ldots, N-1 \)

\( x_{\text{vel}}(k) \leq b + \varepsilon \), \( k = 1, \ldots, N \)

where the first term of the objective function is target tracking criteria that is equivalent to \( (x_{\text{vel}}(t+k|t) - x_{\text{vel}}(t+k|t))^2 Q_k (x_{\text{vel}}(t+k|t) - x_{\text{vel}}(t+k|t)) \)

The second term is the prediction of the future state to ensure that the vehicle stays within the environment boundaries.
and \( Q_i \) is the weight of it, \( x_{sel}(t+k|t) \) is the predicted position of target and is computed by the target trajectory prediction procedure described in the section 2.5. The second term is the set-point weighting criteria with the weight \( R_i \) and is equivalent to \( \Delta u(t+k|t)^T R_i \Delta u(t+k|t) \). \( \Delta u(t+k|t) \) is the predicted set-point increment. The first constraint is the linearization dynamic modelling of the vehicle and the next two are predicted input increment constraints, and predicted set-point constraints, that must be adhered to the vehicle manoeuvrability 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, and hence increases the computational complexity of it. 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 with some available algorithms.

In the previous section the state of the obstacle was assumed a Gaussian random variable and 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 centre of the ellipse is the mean vector of the 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 \( \delta \). Based on the concept of error ellipse [25], the analytical form of this ellipse is given as:

\[
\{ (x, y) : x^T D^{-1} T_{smi} x - k = 0 \} 
\]

where \( T_{smi} \), \( D \) and \( k \) are computed as follows:

\[
T_{smi} = [v_1^T v_2^T]
\]

\[
D = \text{diag}(\lambda_1, \lambda_2)
\]

\[
k = -2 \ln(\delta)
\]

(15)

\( v_1 \) and \( v_2 \) are the eigenvectors corresponding to the eigenvalues \( \lambda_1 \) and \( \lambda_2 \). The semi-axis of the ellipse can be computed as follows:

\[
a = \sqrt{k \times \lambda_1}
\]

\[
b = \sqrt{k \times \lambda_2}
\]

(16)

This 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, meaning 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, yet the non-convexity of the search space is still a problem. The main idea is the linearization of the search space at each time step. To do this, in the first step, a line passing from the gravity centre of vehicle to gravity centre of obstacle, intersected with the expanded error ellipse is drawn. Then in the second step, a tangent line at this intersection point is computed and used as the boundary of the optimization problem 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, \( N \) linear constraints exist for avoiding an obstacle as follows:

\[
A_i x \leq b_i, \quad i = 1, 2 \ldots N
\]

(18)

In the Fig. 2 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.

Moreover, as it is standard in all practical MPC, the slack variable \( \mathcal{E} \) is used to soften the above obstacle
avoidance constraints. Upon adding the term $\rho \epsilon^2$ 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:

$$\min \sum_{k=0}^{N-1} \left[ Q \left( x_{\text{ref}}(t+k|t) - x_q(t+k|t) \right)^2 + R_x^2 \left( u(t+k|t) \right)^2 \right] + \rho \epsilon^2$$

s.t.

$$x_{\text{ref}}(k+1) = A_{\text{ref}} x_q(k) + B_{\text{ref}} u(k), \quad k = 0, \ldots, N-1$$

$$\Delta u_{\text{min}} \leq \Delta u(t+k|t) \leq \Delta u_{\text{max}}, \quad k = 1, \ldots, N$$

$$u_{\text{min}} \leq u(t+k|t) \leq u_{\text{max}}, \quad k = 0, \ldots, N-1$$

$$A_{\text{new}} x_{\text{new}}(k) \leq b + \epsilon, \quad k = 1, \ldots, N$$

(19)

where $\rho$ is the weight coefficient of penalization.

5. Convex Optimization modelling of LTV-MPC

For convex modelling of MPC in Equation (19), it is assumed $u(t) = [u(t), u(t+1), \ldots, u(t+N-1), \epsilon]^T$ is the input vector and $\Delta u(t) = [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-1), \epsilon]^T$ is the input increment vector. Also $x_{\text{ref}}$ is the trajectory prediction vector of the target and $x_q$ is the trajectory prediction of the vehicle that must be computed by the optimization problem. Based on this definition, the problem in Equation (19) can be modelled as follows:

$$\min (X_{\text{ref}} - x_q)^T Q (X_{\text{ref}} - x_q) + \sum_{i=1}^{N} \left[ \Delta u_i^T R_i \Delta u_i + \Delta u_i^T \Delta u_i \right]$$

s.t.

$$x_{\text{ref}} = ST U_i + T_i$$

$$u_{\text{min}} \leq U_i \leq u_{\text{max}}$$

$$\Delta U_{\text{min}} \leq \Delta U_i \leq \Delta U_{\text{max}}$$

$$A_{\text{ref}} x_q \leq b + \epsilon, \quad i = 1, 2, \ldots, N$$

(20)

Where

$$Q = \begin{bmatrix} Q_1 & 0 & 0 & 0 \\ 0 & Q_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Q_N \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & R_N \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} I_{N \times N} \mid 0_{N \times 1} \end{bmatrix}, \quad \Gamma^e = \begin{bmatrix} 0_{N \times (N+1)} \mid I_{N \times 1} \end{bmatrix}$$

$$A = \begin{bmatrix} A_{11} & \ldots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N \times 1} & \ldots & A_{N \times N} \end{bmatrix}, \quad T = \begin{bmatrix} A^2 \\ \vdots \\ A^N \end{bmatrix}$$

$$S = \begin{bmatrix} A^0 & A^1 & \ldots & A^N \end{bmatrix}, \quad \Phi = \begin{bmatrix} A^0 \\ \vdots \\ A^N \end{bmatrix}$$

(21)

This problem can be modelled as a function of the input increment vector $\Delta u_i$ by the variable definition $x_{\text{ref}}(t) = u(t-1)$ and the new state vector is $x_{\text{new}} = [x(t), x_q(t)]^T$ and consequently the new dynamic model of the vehicle has the following form:

$$x_{\text{new}}(t+1) = A_{\text{new}} x_{\text{new}}(t) + B_{\text{new}} \Delta u(t)$$

$$A_{\text{new}} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad B_{\text{new}} = \begin{bmatrix} B \\ I \end{bmatrix}$$

(22)

Also the relation between the new state vector, $x_{\text{new}}$ the state vector of the vehicle $x_{\text{ref}}$ and the input vector $u_i$, is as follows:

$$x_{\text{ref}} = MX_{\text{new}} + GU_i$$

$$u_i = M^T X_{\text{new}}$$

(23)

By substituting the above relation in Equation (20) and simplifying the closed form, the final convex optimization problem can be found as follows:

$$\min \frac{1}{2} \Delta U^T H \Delta U + F^T \Delta U$$

s.t.

$$\Delta U_{\text{min}} \leq \Delta U_i \leq \Delta U_{\text{max}}$$

$$U_{\text{min}} - M^T T x_{\text{new}} \leq M^T S T \Delta U_i \leq U_{\text{max}} - M^T T x_{\text{new}}$$

$$G_i \Delta U_i \leq w_i - P_i x_{\text{new}}$$

(24)

Where the coefficient matrices are computed as follows:

$$H = S M^T Q M S + R \rho$$

$$\Phi = (2x_h T M^T Q M S - 2 x_{tg} T Q M S) \Gamma$$

$$w_i = b_i$$

$$P_i = A_i \Gamma$$

(25)

6. Simulations and Results

The simulation of the simplified method is executed by the CVX toolbox [25] in MATLAB. To start, the initial state of the target, obstacle and vehicle are selected $x_{tg} = [87.1, 0.86, 5.1, 0.05]^T$, $x_h = [85.5, 0.1, 75.5, -5.8, -9]^T$ and $\Gamma = [85.725, 0.08]$. The decision making period is $T = 0.05$ and the covariance matrix of the sensor noise, that is strongly range dependent, is assumed as follows:

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.05 \end{bmatrix}$$

(26)

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

The simulation results are shown in the Fig. 3 and Fig. 4 for two different simulations using CVX toolbox in Matlab. Since the motion of obstacle and target are modelled stochastically, motion of target, obstacle and consequently vehicle must be different for separate simulations. Based on these results, it can be understood that the proposed method is independent of the behaviour of objects in the dynamic environments and can be used for planning of an autonomous vehicle in cluttered environments.

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 modelling of the proposed planning method. Our future work is related to apply this idea to a network of vehicles using decentralized data fusion approaches.

References


