Decentralized Reflective Model Predictive Control of Multiple Flying Robots in Dynamic Environment

David H. Shim H. Jin Kim Shankar Sastry

Department of Electrical Engineering & Computer Sciences
University of California at Berkeley, Berkeley, CA 94720
{hcshim,jin,sastry}@eecs.berkeley.edu

***Abstract*— In this paper, we present a nonlinear model predictive control (NMPC) for flying multiple autonomous helicopters in a complex three-dimensional dynamic environment. The NMPC provides a framework to solve optimal discrete control problem for the nonlinear system under state constraints and input saturation. The uniqueness of our approach lies in the capability to combine the trajectory generation with operational constraints and stabilization of vehicle dynamics by including a potential function reflecting the state information of a possibly moving obstacle or other agents to the cost function. We present various realistic scenarios which show that the integrated approach outperforms a hierarchical structure composed of a separate controller and a path-planner based on the potential function method. Furthermore, the computation load of this approach is light enough to be used for the real-time control of autonomous helicopters.**

I. Introduction

Control and decision-making processes for many complex dynamic systems often consist of many sub-blocks. Control systems for the mobile robots or aircraft are good examples; Each robot or vehicle is often composed of upper-level rule-based logic, and low-level stabilization or actuation layer. Although this hierairchical/horizontal division allows one to approach the complex problem by solving tractable subproblems, the lack of proper coordination between sub-blocks often results in the significant inefficiency or dangerous situation. This paper presents an approach to integrate the path-planning and vehicle stabilization problem for each vehicle and solve it decentrally.

Although the techniques presented in this paper are applicable to various types of mobile robots, we mainly consider radio-controlled model-size helicopters. Autonomous helicopters, or so-called rotorcraft-based unmanned aerial vehicles (RUAVs) shown in Fig. 1 as our experimental platform, have emerged as useful platforms for intelligent mobile robots, due to their flight capabilities.

Nonlinear model predictive control (NMPC) [1] is promising for controlling *nonlinear* systems with operating constraints, such as RUAVs. However, the need for 'fast' control algorithms for many dynamic systems as well as other theoretical issues has constricted the implementation of NMPC. In [2], we applied NMPC for the control of RUAVs in the presence of input and state constraints. The minimization problem was solved with a gradient-descent method as in [3], which is computationally light and fast. The proposed NMPC was employed as an *online* trajectory generation and tracking control layer in a hierarchical flight management system for RUAVs, and outperformed the linear controller in tracking aggressive trajectories under parametric uncertainty [2].

This research was supported by ARO MURI DAAD19-02-1-0383.

Fig. 1. Flight experiment with multiple autonomous helicopters (Photo Courtesy of Hoam Chung)

Planning a collision-free path for a robot to move from an initial to a final configuration is a central problem in robotics and has been the topic of extensive research. However, many versions of this complex problem have been shown PSPACE hard [4] even in a static environment. Also, many methods work only for the discrete state space or for the special shape of obstacles [5]. Deterministic performance guarantee exists only for very simplified cases (for example, see [6] for the two-dimensional aircraft with constant linear speeds and linear constraints using a centralized approach). Potential function methods [7] have dominated the obstacle avoidance research because the idea of generating a repulsive field around each obstacle is intuitive and simple to implement. Although it is well-known that such approaches are prone to local minima and there have been some works on generating so-called navigation functions that are free from local minima [8], generating a navigation function is computationally involved and thus not suitable for many online navigation problems.

In this paper, we extend our previous work to resolve the path planning and optimal control problem for multiple mobile robots in a complex three-dimensional environment by combining ideas from nonlinear model predictive control and potential function techniques. We show that various realistic scenarios involving multiple RUAVs and obstacles in a complex three-dimensional environment can be solved using this integrated approach.

The remaining parts of this paper are organized as follows: Section II presents the mathematical framework including the model helicopter dynamics, the decentralized nonlinear model predictive tracking controller under input saturation and state constraints, and how to utilize the state information of moving obstacles or other agents to avoid collision. Section III presents the simulation results in various realistic examples involving static/moving obstacles and multiple helicopters in a three-dimensional complex environment. Section IV and V conclude the paper with the discussion and future directions.

II. Problem Formulation

This section presents the formal framework for solving the path planning and optimal control problem for multiple RUAVs in a complex three-dimensional environment the nonlinear model predictive control in an integrated manner...

A. Model Predictive Control

In this paper, we consider a system of N (possibly heterogeneous) flying robots in dynamic environment. The dynamics of each vehicle can be described by the following set of controlled differential equations:

$$\dot{\mathbf{x}}_i = f_i(\mathbf{x}_i) + g_i(\mathbf{x}_i)\mathbf{u}_i, \qquad i = 1, \cdots, N, \tag{1}$$

with the initial conditions $x_i(0)$, and where $x_i(t) \in \mathcal{X}_i$ and $u_i(t) \in \mathcal{U}_i$. Each $\mathcal{X}_i$ is a constraint space, subset of $\mathbb{R}^{n_{x_i}}$, and each $\mathcal{U}_i$ is the input space for the vehicle i. As we assume the individual vehicle dynamics are decoupled, we introduce the following vector notation for the overall system;

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{2}$$

where $\mathbf{x} \in \Pi_{i=1}^N \mathcal{X}_i$, and $\mathbf{u} \in \Pi_{i=1}^N \mathcal{U}_i$.

We are interested in solving a *decentralized discrete-time optimal control* problem for the ith subsystem, i.e. to find the optimal input sequence $\{\mathbf{u}_i^*(k)\}_{k=1}^T$ such that

$$\{\mathbf{u}_i^*(k)\}_{k=1}^T = \arg\min \sum_{k=1}^{T} q_i(\mathbf{x}(k), \mathbf{u}(k)) + q_{if}(\mathbf{x}(T+1)) \tag{3}$$

subject to the differential equations (1) and (2), and the terminal constraint $x(T+1) \in \mathbf{X}_f$.

The *nonlinear model predictive control* (NMPC) problem consists of the following steps; 1) solve for the optimal control law starting from the state $\mathbf{x}(k)$ at time k, 2) implement the oprimal input $\mathbf{u}^*(k), \cdots, \mathbf{u}^*(k+\tau-1)$ for $1 \le \tau \le T$[1], and 3) repeat these steps from the state $\mathbf{x}(k+\tau)$ at time $k+\tau$.

Often the quadratic function is used for $q_i(\mathbf{x}(k), \mathbf{u}(k))$ and $q_{if}(\mathbf{x}(T+1))$, and see [3], [2], for a nonlinear-programming algorithm using Lagrange multipliers.

B. Model Helicopter Dynamics

Although the techniques presented in this paper are applicable to varioustypes of mobile robots, we mainly consider autonomous helicopters as our experimental platform. Multi-input multi-output (MIMO), nonlinear characteristics, severe disturbance, and a wide flight envelope present difficulty in controlling autonomous helicopters. The helicopter is an under-actuated system, whose configuration space is $SE(3) \triangleq \mathbb{R}^3 \times SO(3)$ yet only four degree-of-freedom can be achieved by four inputs to the lateral cyclic pitch, longitudinal cyclic pitch, main rotor collective pitch, and tail rotor collective pitch [9][2].

Let the superscripts S and B denote spatial and body coordinate, ϕ, θ, and ψ denote roll, pitch, and yaw, and p, q, and r are their rates, respectively. The overall system dynamics are divided into the kinematics (Eqn. (4)) and the system-specific dynamics (Eqn. (5)) denoted by the superscripts D and K[3]:

$$[\dot{x}^S, \dot{y}^S, \dot{y}^S] = R^{B\to S}[\dot{x}^B, \dot{y}^B, \dot{z}^B] \tag{4}$$

$$\dot{\mathbf{x}}_i^D(t) = f_c(\mathbf{x}_i^D(t), \mathbf{u}_i(t)) \tag{5}$$

$$\begin{aligned} \mathbf{x}_i &= [\mathbf{x}_i^K, \mathbf{x}_i^D] \in \mathbb{R}^{n_x} \\ \mathbf{x}_i^K &= [x^S, y^S, z^S, \phi, \theta, \psi] \\ \mathbf{x}_i^D &= [u, v, w, p, q, r, a_{1s}, b_{1s}, r_{fb}] \\ \mathbf{u}_i &= [u_{a1s}, u_{b1s}, u_{\theta_M}, u_{\theta_T}] \in \mathbb{R}^{n_u}, \end{aligned}$$

where a_{1s} and b_{1s} are longitudinal and lateral flapping angles, and r_{fb} is the feedback gyro system state. A transformation matrix between the spatial and body velocities

[1] In all examples we present, we set $\tau = 1$

[2] We exclude the rotor throttle from our defi nition of control inputs, since we assume that the rotor rpm is maintained constant by an engine governor.

[3] For the notational simplicity, we occasionally drop the subscript i to denote the i_th helicotper.

is given by $R^{B\to S} \in SO(3)$, i.e., the rotational matrix of the body axis relative to the spatial axis, represented by ZYX Euler angles $[\phi, \theta, \psi]$. Newton-Euler equation yields the differential equation (5) for $\mathbf{x}^D$, which is characterized by nonlinear functions of the force and moment terms [9].

After approximating force and moment terms under the low-velocity/small-attitude assumption, the overall dynamics can be written in the affine form shown in the Eqn. (6).

C. Trajectory Generation and Tracking under Input/State Constraints

We use the following quadratic functions of state variables and inputs as the cost for the ith helicopter at time t in Eqn. (3),

$$\begin{aligned} q_i(\mathbf{x}(k), \mathbf{u}(k)) &= J_i^{st}(\mathbf{x}_i(k), \mathbf{u}_i(k)) \quad (7)\\ &\triangleq \frac{1}{2}\left(\mathbf{y}_{id}(k) - C\mathbf{x}_i(t)\right)^T Q \left(\mathbf{y}_{id}(t) - C\mathbf{x}_i(t)\right)\\ &+ \frac{1}{2}\mathbf{u}_i(k)^T R \mathbf{u}_i(k) \;,\\ q_{if}(\mathbf{x}(T+1)) &\\ &= \frac{1}{2}\left(\mathbf{y}_{id}(T{+}1) - C\mathbf{x}_i(T{+}1)\right)^T P_0(\mathbf{y}_{id}\,(T{+}1) - C\mathbf{x}_i(T{+}1)) \end{aligned}$$

where $\mathbf{y}_{id}(\cdot)$ denotes the desired trajectory for the ith helicopter.

In order to generate physically realizable trajectories, input constraints are enforced by projecting each $\mathbf{u}_k$ into the constraint set. In our helicopter model, this corresponds to $[u_{a1s}, u_{b1s}, u_{\theta_M}, u_{\theta_T}] \in [-1, 1]^4$. State constraints are also incorporated as an additional penalty in the cost function, i.e., the cost for the ith helicopter at time t, $q_i(\mathbf{x}(k), \mathbf{u}(k))$ of Eqn. (3), now includes

$$J_i^{sc}(\mathbf{x}_i(k)) \triangleq \sum_{l=1}^{n_x} S_{il} \max(0, \mid x_{i,l}(k) - x_{i,l}^{\text{sat}} \mid)^2 \;, \quad (8)$$

where $x_{i,l}(k)$ denotes the ith state variable for the ith helicopter at time t, and S_{il}, and $x_{i,l}$ are constants.

D. Decentralized collision-free trajectory generation

Our model-predictive path planning strategy adopts the idea from the potential field method [7], [4], which has been popular in path planning for mobile robots.

The cost (3) can be formulated to reflect the aspect of a potential function for path planning in the environment with moving obstacles or other agents. This allows the trajectory generation and vehicle stabilization to be combined into a single problem. In this scenario, we assume that each vehicle is aware of other vehicles' real-time location via some type of communication channel and solve the optimization decentrally.

The following potential function term is added to the cost function for the helicopter j, whose position at time t is denoted by $\mathbf{x}_k^j$.

$$\begin{aligned} &J_i^{moa}(\mathbf{x}_j(k)) \quad (9)\\ &= \sum_{l\neq j}^{N} \frac{K_{jl}}{a_j^2(x_j(k)-x_l(k))^2 + b_j^2(y_j(k)-y_l(k))^2 + (z_j(k)-z_l(k))^2}, \end{aligned}$$

where $(x_l(k), y_l(k), z_l(k))$ denote the position of the helicopter l at time k, and constants a_j, b_j and K_{jl} determine the shape of repulsive potential field and thus, the adjusted trajectory. We add a repulsive potential of the same form for the moving obstacles if the environment is dynamic. Thus, the information of the other helicopters or moving obstacles, such as their position and velocity, if known, is used in *predicting* the cost over the next T horizon. In fact, this aspect attributes to the good performance of our approach over the conventional potential-function based method as discussed in Section III.

E. Three-dimensional pursuit-evasion game

In this case, we consider two UAVs in a close-range air operation. One UAV is in pursuit of the other UAV. The pursuing UAV's goal is to align its heading to the target UAV and reduce the distance without crashing into the target.On the other hand, the other UAV tries to escape from the point where the chance for damage is minimal. This situation is analogous to "grab the tail of the target" and try to "shake off the enemy from its tail", i.e. UAV A locates itself to maximize the chance to attack the target while minimizing the risk of being exposed to UAV B's fire. This can be formulated as two separate objectives, which are not necessarily in conflict: 1) align its heading to the target vector and 2) avoid being aligned in the target's heading.

In Fig. 2, the relationship among the relative distance and headings in the cost functions for pursuit-evasion is shown. For purser, the deviation of the relative heading angle between $\mathbf{x}_P^H \triangleq R_P^{B\to S}[1,0,0]^T$ and the relative position vector of unit length $\mathbf{x}_{E\leftarrow P}$ is penalized to obtain the best aim. For evaders, the cost function as a function of the same angle α_D is formulated to become zero when $\alpha_D = \pm 90°$ and maximum at 0° and 180°. In addition to these penalties on headings, the relative distance between the pursuer and the evader needs to be included as a part of the cost function for the pursuer and the evader. For pursuer, the distance between these two are penalized for more effective pursuit. The relative distance is inversely penalized in the cost function of the evader for the exactly opposite reason.

The following cost functions are proposed for a pursuer and evader, respectively;

$$\begin{aligned} &J_p(\mathbf{x}_P(k), \mathbf{x}_E(k)) \quad (10)\\ &= K_p \mathbf{x}_{E\leftarrow P}^T \left(K_{dist} I - \text{sgn}(\mathbf{x}_{E\leftarrow P} \cdot \mathbf{x}_P^H)\mathbf{x}_P^H \cdot (\mathbf{x}_P^H)^T\right) \mathbf{x}_{E\leftarrow P}\\ &J^e(\mathbf{x}_P(k), \mathbf{x}_E(k)) \quad (11)\\ &= K_e \mathbf{x}_{E\leftarrow P}^T \left(\mathbf{x}_P^H \cdot (\mathbf{x}_P^H)^T\right) \mathbf{x}_{E\leftarrow P} \;. \end{aligned}$$

$$
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{p} \\ \dot{\theta} \\ \dot{q} \\ \dot{\psi} \\ \dot{r} \\ \dot{a_{1s}} \\ \dot{b_{1s}} \end{bmatrix} = \begin{bmatrix} -g\theta - \frac{c_2}{m}a_{1s} + vr - wq \\ g\phi - \frac{c_2}{m}b_{1s} - \frac{c_6}{m} - ur + wp \\ g - \frac{c_2}{m} + uq - vp \\ p + q\sin\phi\tan\theta + r\cos\phi\tan\theta r \\ -\frac{c_4}{I_{xx}}a_{1s} - \frac{c_2h_M + c_9}{I_{xx}}b_{1s} - \frac{c_2y_M + c_6h_T}{I_{xx}} + \frac{I_{yy}-I_{zz}}{I_{xx}}qr \\ q\cos\phi - r\sin\phi \\ \frac{c_2h_M + c_{10}}{I_{yy}}a_{1s} - \frac{c_4}{I_{yy}}b_{1s} + \frac{c_2l_M - c_8}{I_{yy}} + \frac{I_{zz}-I_{xx}}{I_{yy}}pr \\ q\sin\phi\cos\theta + r\cos\phi\cos\theta \\ \frac{c_2l_M}{I_{zz}}b_{1s} + \frac{-c_4 + c_6l_T}{I_{zz}} + \frac{I_{xx}-I_{yy}}{I_{zz}}pq \\ -q - \frac{1}{\tau_f}a_{1s} + A_{b_{1s}}b_{1s} \\ -p + B_{b_{1s}}a_{1s} - \frac{1}{\tau_f}b_{1s} \end{bmatrix}
$$

$$
+ \begin{bmatrix} -\frac{c_1}{m}a_{1s} & 0 & 0 & 0 \\ -\frac{c_1}{m}b_{1s} & -\frac{c_5}{m} & 0 & 0 \\ -\frac{c_1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\left(\frac{c_3}{I_{xx}}a_{1s} + \frac{c_1h_M}{I_{xx}}b_{1s} + \frac{c_1y_M}{I_{xx}}\right) & -\frac{c_5h_T}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \left(\frac{c_1h_M}{I_{yy}}a_{1s} - \frac{c_3}{I_{yy}}b_{1s} + \frac{c_1l_M}{I_{yy}}\right) & -\frac{c_7}{I_{yy}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \left(\frac{c_1l_M}{I_{zz}}b_{1s} - \frac{c_3}{I_{zz}}\right) & \frac{c_5l_T}{I_{zz}} & 0 & 0 \\ 0 & 0 & A_{u_{a_{1s}}} & A_{u_{b_{1s}}} \\ 0 & 0 & B_{u_{a_{1s}}} & B_{u_{b_{1s}}} \end{bmatrix} \begin{bmatrix} u_{\theta_M} \\ u_{\theta_T} \\ u_{a_{1s}} \\ u_{b_{1s}} \end{bmatrix} \quad (6)
$$

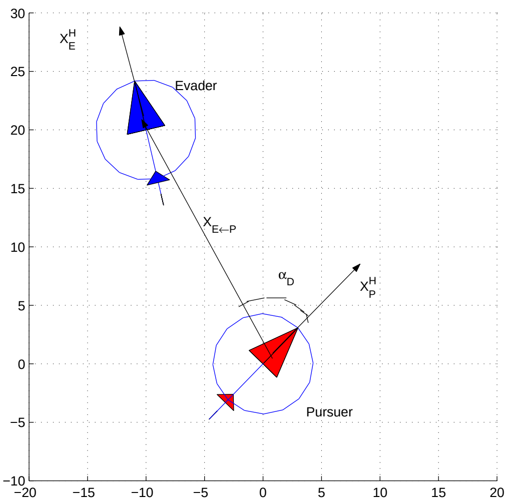

Fig. 2. pursuit-evasion, where a pursuer wants to minimize the relative angle ...

The overall cost function term for each player is now given by

$$
\begin{aligned}
&q(x(t), u(t)) \qquad (12)\\
&\quad = J^{st}(x(t), u(t)) + J^{sc}(x(t), u(t)) \\
&\quad + J^{bdry}(x(t), u(t)) + J^{moa}(x(t), u(t)) \\
&\quad + J^{p}(x(t), u(t)) + J^{e}(x(t), u(t)) \;.
\end{aligned}
$$

In addition to the two usual cost functions J^{st} and J^{sc}, J^{bdry}, J^{oa}, J^{p} and J^{e} are included in (1). J^{bdry} is the cost function which enforces the UAVs stay in the predefined region. If this function is not included, the evading agent may fly indefinitely away in any direction to flee from the pursuer. J^{moa} is the cost function for preventing collisions between two agents: without this, the penalty on the mutual distance in J^{p} may lead the pursuer to crash into the other agent. The cost functions for pursuit and evasion may be introduced together as a sum of pursuing part and evading part. By combining these two, one agent may have a variable combination of pursuit and evasion. Two extreme cases are pure pursuit and evasion. A pursuer persistently follows the target even if it is exposed to a higher risk from the other agent. The pure evader only avoids being aligned along the pursuing agent's line of sight. One may find a satisfying combination of these two traits as shown in Section III-D.

III. Simulation Results

In this section, we evaluate the effectiveness of the non-linear model predictive trajectory planning and tracking controller proposed in Section II in various scenarios.

A. Collision-avoidance Planning and Control under Input/State Constraints

In this example, five helicopters are originally given straight-line trajectories that will lead to a mid-air collision at $(50, 0, 33)$ ft, as shown in Fig. 3(a). The potential function in Eqn. (9) is added into the cost function of each helicopter, to replan the trajectory. In order to generate the plausible control input while trying to avoid the collision, the input saturation conditions were enforced. Also included in the cost function is the state constraints in the form of Eqn. (8), with $[\phi_{\text{sat}}, \theta_{\text{sat}}, u_{\text{sat}}, v_{\text{sat}}, w_{\text{sat}}, p_{\text{sat}}, q_{\text{sat}}, r_{\text{sat}}] = [\pi/6, \pi/6, 16.7, 16.7, 16.7 \text{ ft/s}, \pi/2, \pi/2, \pi/3 \text{ rad/s}]$ in order to contain the overall vehicle response at an acceptable range. Fig. 3(b) shows the resulting trajectories that each helicopter actually achieved.

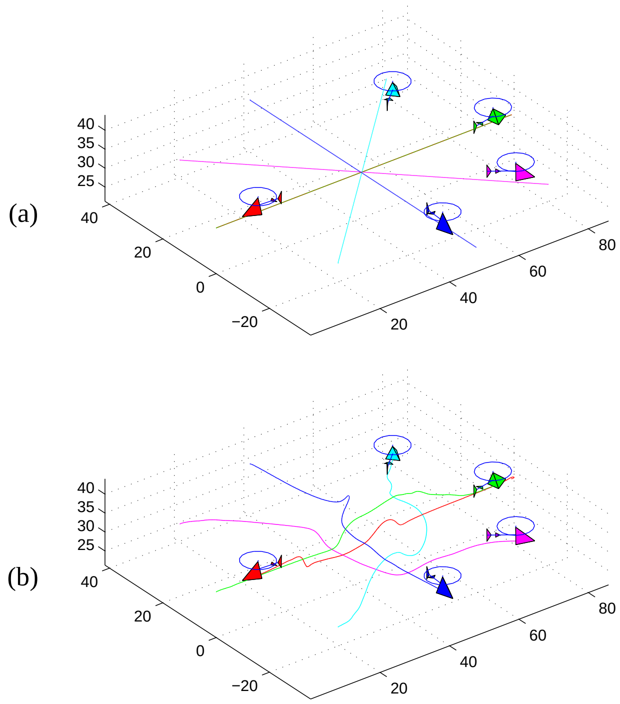

Fig. 3. Trajectory of five helicopters: (a) initial configuration and the destination of each helicopter, (b) their trajectories executed

The result shown in Fig. 4 shows that our integrated approach outperforms the purely potential-function method, when te potential function is employed as a path planning layer separate from a vehicle stabilization layer. The helicopter heading to the left is controlled by refelctive model predictive control, whereas the helicopter flying to the right, assumed to have already stabilized linear dynamics, uses the potential function method as a trajectory generation layer. When the potential function is considered in generating the trajectory only for each time step, it caused severe chattering (Point A), slow convergence back to the desired trajectory (Point B) and overshoot (Point C) in the vehicle response. On the other hand, the nmpc-based method resulted in smooth trajectory without overshoot.

B. Flying in dynamic, unknown environment

In the following example, we demonstrate the feasibility of the NMPC framework as a trajectory planner in a dynamic, unknown three-dimensional environment. In this scenario, the vehicles are requested to fly through a narrow channel, without colliding into walls or other vehicles, where obstacles appear at unknown time. As shown in Fig. 5(a), a team of three helicopters are commanded to fly to the points that are 100 ft ahead in formation keeping the minimum separation for the safety. In Fig. 5(b), the vehicles 1 and 2 changes their trajectory to pass through an alley between two walls when they detect the wall, and the three helicopters go into hover when a popup wall suddenly appears and blocks their flight path. In Fig. 5(c), the helicopters resume forward flight and the helicopter 3 lowers the altitude to avoid a popup obstacle which suddenly appeared. In (d), the helicopters circumvent a column, each choosing the optimal direction that minimizes the deviation from their straight-line path. Again, each figure shows the actual trajectories followed by the nonlinear helicopter dynamics with the online NMPC module.

C. Flying in a complex environment

In this example, we apply the NMPC framework for a navigation problem in which a UAV is requested to fly through a space full of buidling-like obstacles. This type of situation often arises in a urban area, filled with buildings of irregular sizes.

For this simulation, a realistic three-dimensional map filled with buildings of random width and height is generated as shown in Fig. 6(a). Then a UAV is requested to fly from the rooftop of the building at lower-left on the map to the building at the diagonal side. The NMPC-based flight controller is only given *a priori* with an simple straight flight path that directly connects the start and the destination point. The given trajectory intersects with a number of buildings along the path. The vehicle resolves the collision by maintaining the safe distance from the nearest point from nearby buildings as it travels. The distance to the closest building from the UAV is used to compute the potential function in Eqn. (9) at every position along the finite horizon during the optimization. The process of finding the closest point may be implemented with sensing with laser or ultrasonic sensor system. Fig. 6(a) and (b) shows the output of a simulation. The short lines from the UAV's trajectory to the nearby buildings indicate the vector of the minimum distance from the scanning process. Along the path, the UAV encounters a number of buildings which are too close to itself. The UAV succssfully reachs the destination by detouring the building along the given path.

also by considering k nearest objects not just one...

Fig. 7 shows the result when using a potential function method, i.e.. local minima...roblem

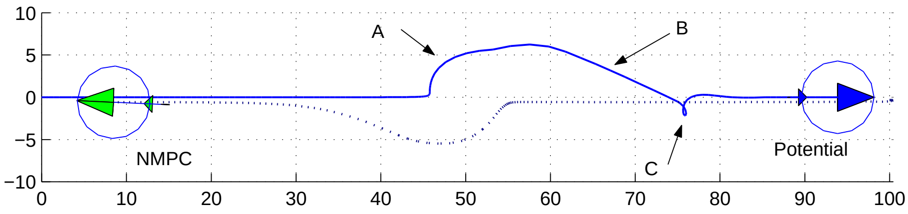

Fig. 4. Model predictive control vs. potential-function approach

D. Three-dimensional pursuit-evasion game

In this example, we consider two UAVs in a close-range air operation. Suppose that one UAV is in pursuit of the other UAV. The pursuing UAV's goal is to align its heading to the target UAV and reduce the distance without crashing into the target. The evading UAV's goal is to stay out of the line of sight of the pursuer as explained in Section II-E.

In Fig. 8, a snapshot of a pursuit-evasion for six seconds interval between two full-time opposite-trait agents are shown. The pursuer attempts to shorten the distance and align its heading to the direction where the other agent is located simulatneously. The evader on the other hand moves away from the pursuer while refraining to show its tail to the pursuing agent as the cost function in Eqn. (12) dictates.

In Fig. 9, a fourteen-second duration of simulation result of two agents with both pursuit and evasion capabilities is shown. The distinction from the previous case is that each agent tends to make more abrupt changes in the flight path to avoid being in the targeting area. At the same time, it attempts to position itself at a more advantageous point whenever and wherever possible.

E. Combining all together: pursuit-evasion game in a confined space with urban obstacles

This exmple, shown in Fig. 10, combined obstacle-avoidance and pursuit-evasion feature as well as the constraints on the magnitude of state variables and inputs. The UAVs in this environment are performing both pursuit and evasion while avoing collision into the buildings in the confined area. As is in the scenario in Section III-C, the UAVs are confined by J^{bndr}.

IV. Implementational Issues and Future work

Horizon length N, number of iterations during optimization, as well as the weighting matrices P_0, Q, and R, are important design parameters related to the simulation speed and closed-loop stability. Step-size Δ_k should be carefully chosen to consistently reduce the cost during the iteration. The selection of Δ_k is also related with the horizon length N as well as other weightings. We selected $N = 25 \sim 30$, and $\Delta_k = 0.0001 \sim 0.001$ for the simulations presented in this paper. The effect of tuning these parameters and adjustment of potential function to also reflect the type, speed or heading of objects require further investigation.

We initialized the initial control sequence by the output of PID controller designed for the linearized model in [9]. By initializing $\mathbf{u}_k$ at the beginning of the optimization at each time step with the $\mathbf{u}_k$ of the previous time sample, the iteration count reduces significantly.

Our NMPC algorithm is written in CMEX format for enhanced computation speed and displayed in Matlab. The simulation ran faster than real time on 1 GHz PC in all the examples shown in this section. Table I summarizes the computation time for the examples shown in Sections III-A – III-E. All the examples were run in a decentralized manner for each helicopter but on a *single* Pentium-III 1GHz notebook. Overall, the proposed NMPC algorithm is a very promising approach for UAV flight control systems where the host vehicle is demanded to operate in a complicated environment.

Although deliberation of the optimality over the time horizon T renders our NMPC-based approach less prone to the local minima than purely potential-function-based methods, we would need T to be large enough to *foresee* over the local minima. Since the computation time directly depends on T, it would be more efficient to combine some high-level logic to in very general settings, rather than blindly increasing T.

V. Conclusion

In this paper, we have formulated a generalized nonlinear model predictive control (NMPC) framework to solve various control/trajectory generation problems involving collision avoidance in a complex three-dimensional space with a large number of moving obstacles and autonomous vehicles. We implemented the proposed NMPC algorithm as an online decentralized optimization controller and evaluated in a wide variety of realistic scenarios. In the examples shown in this paper, the potential function and input/state constraints were included into the cost, and the input saturation conditions were enforced to show the viability of our approach in integrating a trajectory generation layer and a vehicle stabilization layer. Although there exists no guarantee of the uniqueness of the minima, we observed that our approach is less prone to the local

(a)

(b)

(c)

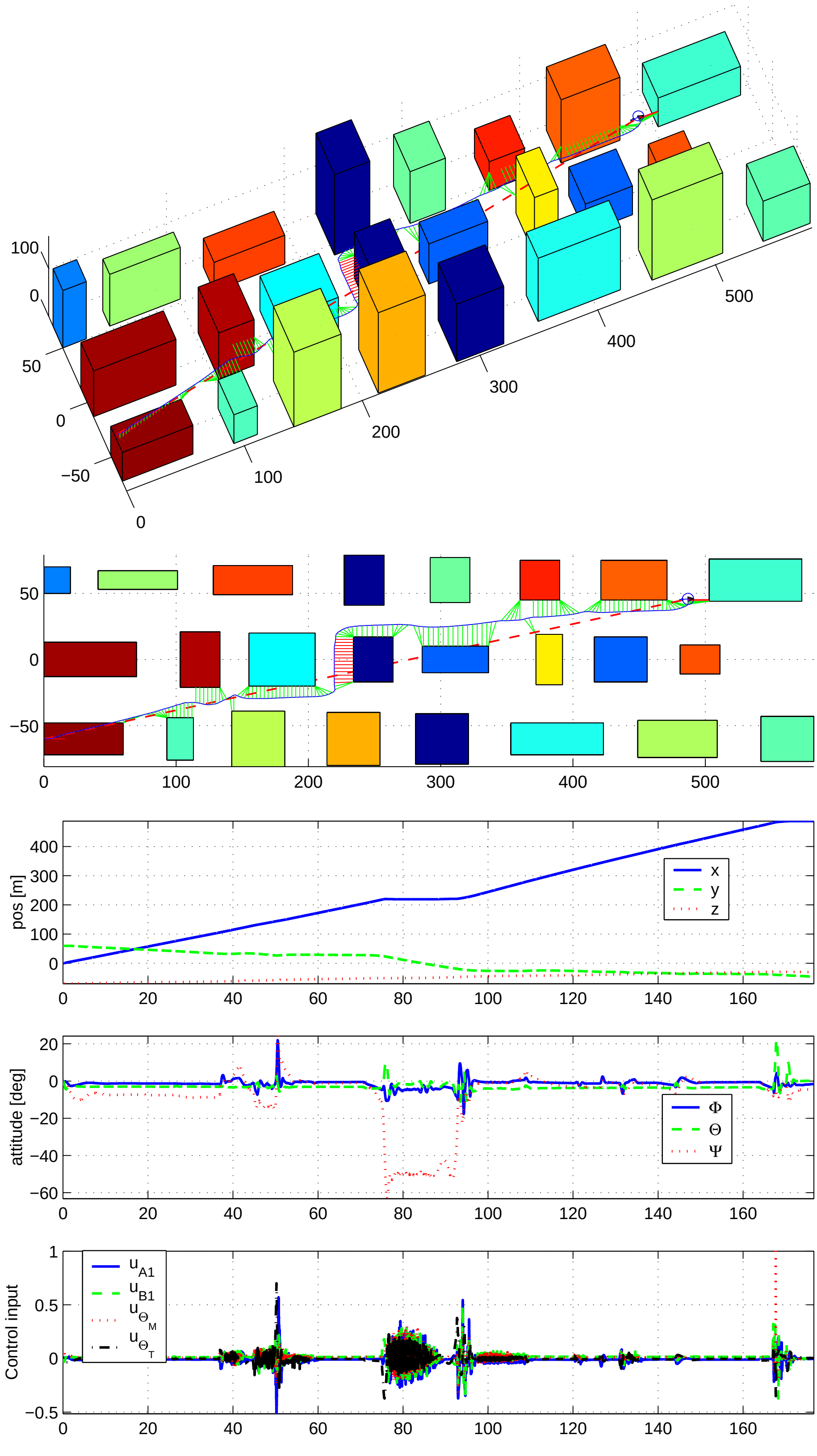

Fig. 6. NMPC in a complex 3-D environment

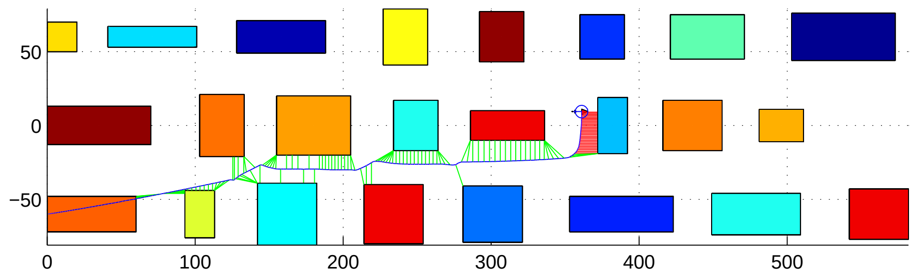

Fig. 7. Potential-function approach in a complex 3-D environment:local minima.

Example	Flight time (sec)	Computation time (sec)	Number of helicopters
A	24	41	5
B	50	29	3
C	176	110	1
D	206	168	2
E	200	173	2

TABLE I

COMPUTATION TIME FOR THE EXAMPLES SHOWN IN SECTION III, RUN ON A *single* NOTEBOOK WITH A PENTIUM-III 1GHZ PROCESSOR, RUNNING ON WINDOWS-XP

minima than the conventional potential-function methods due to the longer-term preview. The computational load of our NMPC formulation using the gradient-descent or conjugate gradient method is low enough to be used for controlling RUAVs real-time.

VI. REFERENCES

[1] F. Allgöwer and A. Zheng, editors. *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*. Birkhäuser Verlag, Basel-Boston-Berlin, 2000.

[2] H.J. Kim, D.H. Shim, and S. Sastry. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *American Control Conference*, May 2002.

[3] G.J. Sutton and R.R. Bitmead. Computational implementation of NMPC to nonlinear submarine. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, volume 26, pages 461–471. Birkhäuser, 2000.

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

[5] A. Zelinsky. A mobile robot navigation exploration algorithm. *IEEE Transactions of Robotics and Automation*, 8(6):707–717, 1992.

[6] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference*, Anchorage, AK, May 2002.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[8] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, Oct. 1992.

[9] D.H. Shim. *Hierarchical Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*. PhD thesis, University of California at Berkeley, 2000.

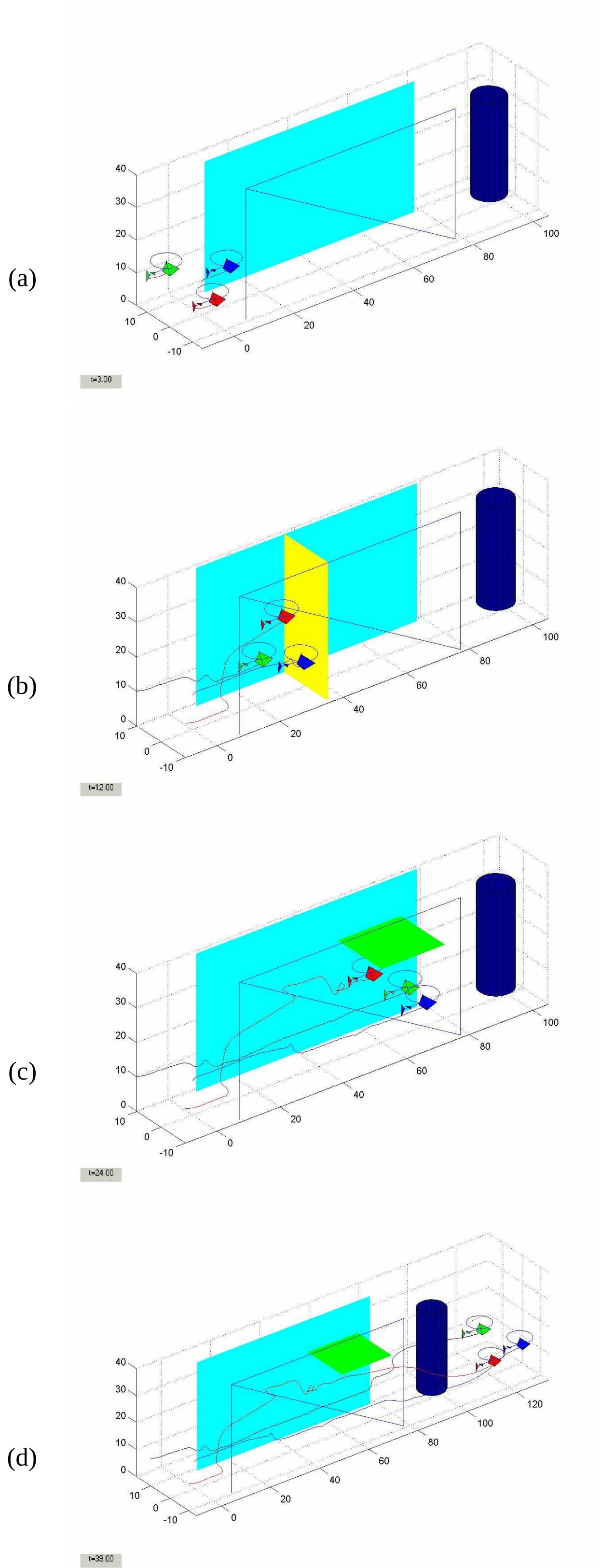

Fig. 5. Loose formation flight of three helicopters in an unknown, dynamic environment (a) their initial positions, (b) avoiding walls, (c) popup obstacles from the ceiling, and (d) column-shaped opstacles.

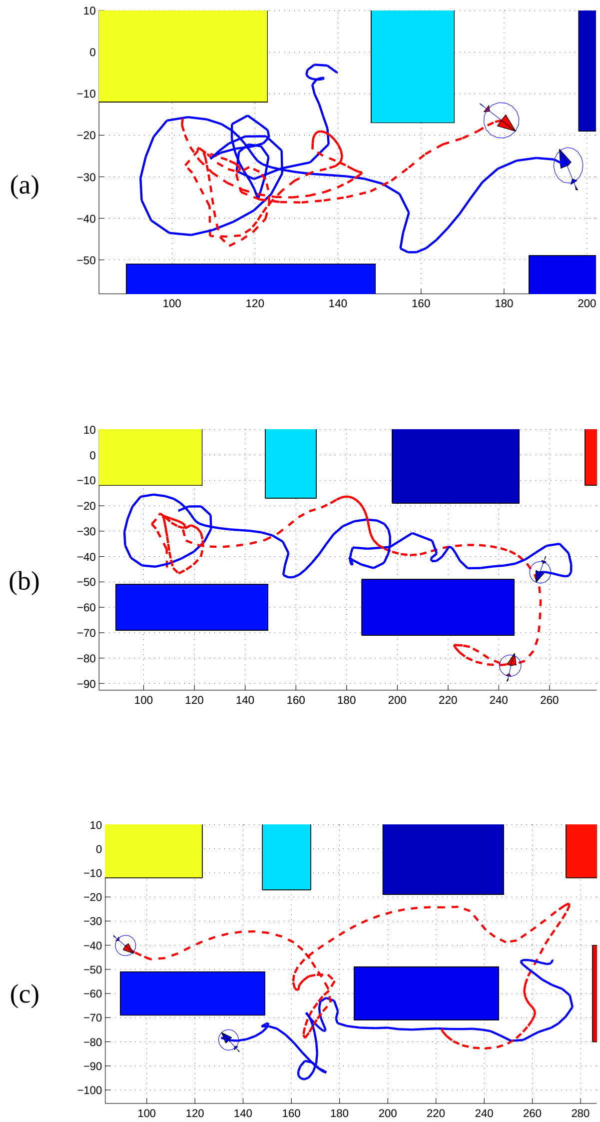

Fig. 10. Top-down view during a symmetric pursuit-evasion game in a complex three-dimensinal environment: both players are pursuing and evading at the same time

(a)

(b)

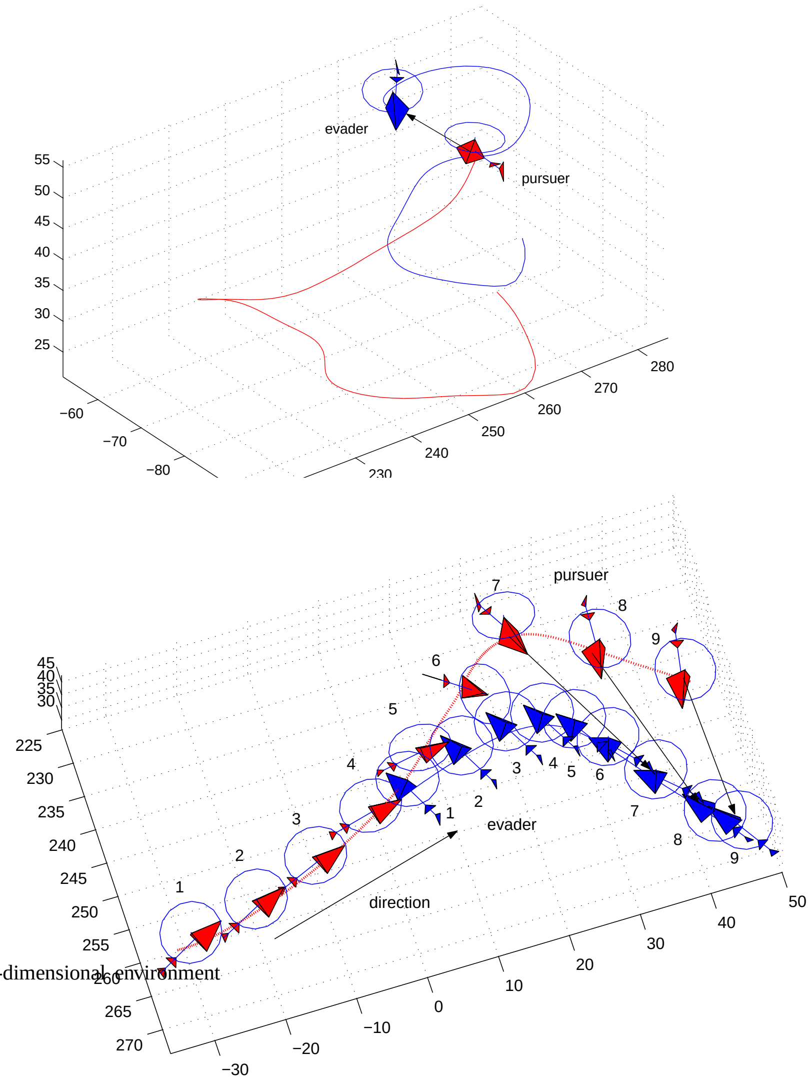

Fig. 8. Pursuit-evasion in a three-dimensional environment

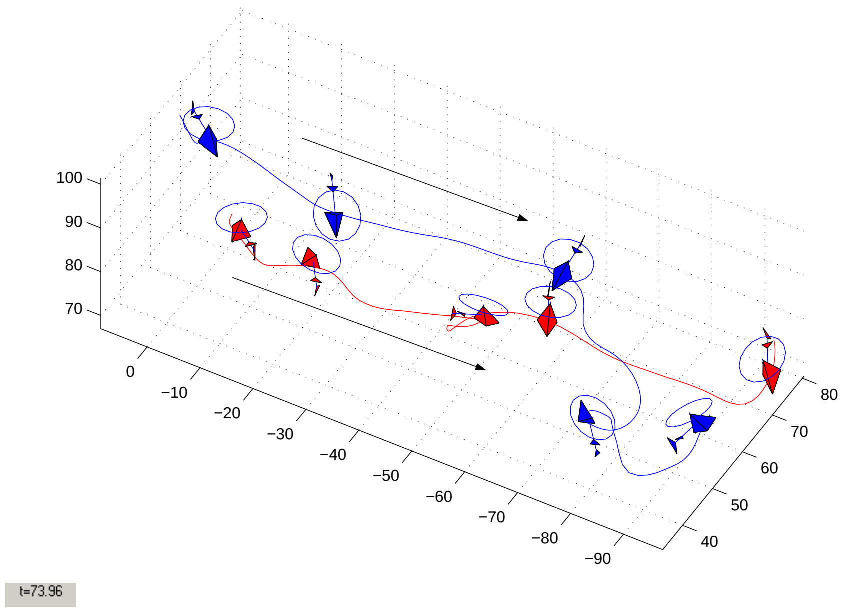

Fig. 9. Symmetric pursuit-evasion in a three-dimensional environment, in which both players are pursuing and evading at the same time.