2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Collision-Free Trajectory Generation of Robotic Manipulators Using Receding Horizon Strategy

Hoam Chung and Soo Jeon

Abstract— **The main objective of this paper is to study the feasibility of the receding horizon (RH) strategy to generate the collision-free on-line optimal trajectory for articulated manipulators under dynamic environments. Firstly, we employ the elliptical set to represent the no-collision zone around each link, and explicitly formulate collision avoidance constraints using state-space inequalities. Secondly, to address the major drawback of intensive computation load, we adopt so called the external active-set strategy that is recently developed by Chung and Polak [1]. Simulation results with two-link robot are presented to demonstrate how to implement the proposed method. Main features and related issues for the feasibility of the RH strategy are discussed in detail.**

I. INTRODUCTION

Optimization-based approach has long been pursued in the operation of articulated manipulators as a means to realize the best possible control law or trajectory for a desired task. Nevertheless, the current practice is still far from the optimal approach and are often limited to the simple form of tracking control using the approximated SISO (Single-Input-Single-output) linear model. The major difficulty in the optimal motion planning of manipulators lies in the complexity of the nonlinear equation describing the multibody dynamics. In addition, its practical application often requires the imposition of the constrains such as the torque limit and the collision avoidance.

The initial attempt in the optimal control of manipulators [2] sought for the solution to the Hamilton-Jacobi equation associated with the unconstrained quadratic optimal control problem. Similar results [3], [4] also appeared in the $\mathcal{H}_\infty$ optimal control problem, which was mainly concerned about the analytic solution without constraints. One of the representative studies in the robot motion planning through the unconstrained nonlinear optimization is reported by Bobrow et al. [5], which considered the off-line optimal trajectory generation strategy that minimizes various physical criteria such as the energy, control effort, etc. If the constraints to be imposed are dynamically changing through the operation, the optimization problem requires running the solver on-line at each time step, the method of which is known as the receding horizon (RH) strategy or the model predictive control (MPC). Specifically to the articulated manipulation, the need for the on-line optimization may arise from some cases where the manipulator is operated in the dynamically changing (unknown) environment, e.g. the collision avoidance for moving obstacles and the human-robot cooperation.

Compared to the historical popularity of RH strategy in the process control [6] and some of its recent successes in vehicular robots [7], the practical application of RH strategy for articulated manipulators is still rare. Among recent approaches are the MPC for the linearized manipulator model [8], [9], the neural network approximation of the manipulator dynamics [10], and the stability guaranteed unconstrained RH control for a direct drive robot [11] using a carefully chosen terminal cost function. It should be noted that, in the majority of the cases, the constraints on the state variables or the input torques are not explicitly considered. As mentioned earlier, one of major advantages of RH-based trajectory generation is that it can handle the hard constraints in state and control variables. Another distinctive feature is that it does not suffer from the local minima arising from the potential field approach [7]. Furthermore, since the RH-based solution always searches for the dynamically and kinematically feasible solution without regard to manipulator configurations, it can be free from the singularity considerations, the trouble of which is encountered in most of traditional motion planning strategies based on task space control. In spite of such attractive features, the intensive computation of the on-line optimization has been a critical issue thus preventing it from being used in practical applications. Apparently, the general nonlinear optimization has not yet been fully implemented in the application of RH-based motion planning for articulated manipulation.

This paper presents the feasibility study on the general nonlinear finite horizon optimal control problem (FHOCP) applied to the articulated manipulation considering the dynamic obstacle avoidance and actuator limits as hard constraints. Firstly, we employ the elliptical set to represent the no-collision zone around each link, and explicitly formulate collision avoidance constraints using state-space inequalities. Secondly, to address the major drawback of intensive computation load, we adopt so called the external active-set strategy that is recently developed by Chung and Polak [1]. Optimal control problems with state space constraints are usually solved by discretizing the dynamics, which results in the conversion of the continuous-time optimal control problem into a discrete-time optimal control problem with many inequality constraints. The central idea of the external active-set strategy is based on the observation that most of those discrete-time inequality constraints are inactive, and may be excluded from the computation if a proper algorithm can generate optimal solutions without including them in the

H. Chung is with Department of Mechanical and Aerospace Engineering at the Monash University, Victoria 3800 Australia. S. Jeon is with Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON Canada N2L 3G1 Hoam.Chung@monash.edu, soojeon@uwaterloo.ca

978-1-4577-0079-8/11/$26.00 ©2011 AACC

computation. It has been reported that the external active-set can dramatically reduce the computation time, in some cases, to a factor of over 100 [1].

This paper is organized as follows. In Section II, we present the formulation of the FHOCP problem and RH-based trajectory generation algorithm. Section III presents the application of the proposed formulation to a two-link manipulator as an example for the feasibility study. In particular, Section III-B illustrates how the external active-set strategy reduces the computational load. Numerical results are provided in Section IV, and our concluding remarks and future works are given in Section V.

II. Formulation

Consider an n link serial manipulator with a joint variable space $Q \in \mathbb{R}^n$. Let $W \in SE(3)$[1] denote its Cartesian workspace and a smooth map $\mathcal{K} : Q \to W$ denote the forward kinematics of the manipulator. Assume that its inverse, $\mathcal{K}^{-1} : W \to Q$, exists and also smooth. We call $q \in Q$ as a joint vector, and a generalized force vector $\tau \in \mathbb{R}^n$ as a joint torque vector. The dynamics of a manipulator can be written in the form of

$$\tau(t) = M(q)\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + g(q(t)), \quad (1)$$

where $q(t)$, $\dot{q}(t)$, and $\ddot{q}(t)$ are the joint position vector, the joint velocity vector, and the joint acceleration vector respectively. $M(q) \in \mathbb{R}^{n\times n}$ is the non-singular mass matrix, $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ is a vector representing Coriolis and centrifugal forces, and $g(q) \in \mathbb{R}^n$ is a vector representing gravitational forces [12]. Note that the above dynamics can always be converted into a nonlinear state-space representation

$$\dot{x}(t) = f(x(t), u(t)) \quad (2)$$

with

$$x(t) \triangleq \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}, \; u(t) \triangleq \tau(t),$$
$$f(x(t), u(t)) \triangleq \begin{bmatrix} \dot{q}(t) \\ M^{-1}(q(t))\Big(u(t) - C(q(t), \dot{q}(t))\dot{q}(t) \\ \qquad\qquad - g(q(t))\Big) \end{bmatrix}. \quad (3)$$

Suppose that we have a continuous desired end effector trajectory represented by $G_d(t)$ with $G_d : t \to W$ for $t \in [0, T]$ and $N_o(t)$ obstacle points in W represented by $d_j(t) = \big(d_j^x(t), d_j^y(t), d_j^z(t)\big) \in \mathbb{R}^3$, $j \in \mathcal{N}_o(t) \triangleq \{1, 2, \ldots, N_o(t)\}$. $N_o(t)$ is assumed to be dependent on time, i.e. the number of obstacle points vary with time. Let us define another time-varying set

$$\mathcal{B}_l(t) \triangleq \big\{p \,|p \in \mathbb{R}^3, \; B_l(p, q(t)) \le 0\big\}, \quad (4)$$

which represents the closed convex region in $\mathbb{R}^3$ such that the l^{th} link of the manipulator is fully contained in it.

[1]Special Euclidean group, $\mathbb{R}^3 \times SO(3)$

The finite-horizon optimal control problem (FHOCP) for the collision free trajectory generation for $t \in [t_i, t_i + h]$ is defined as follows.

$$\min_{u(t)\in\mathcal{U}} J(u(t)), \; t \in [t_i, t_i + h] \quad (5)$$

where $\mathcal{U}$ is a set of Lebesgue square integrable functions with maximum and minimum bounds defined in $t \in [t_i, t_i + h]$. h is the prediction interval of the receding horizon scheme. The FHOCP in (5) is subject to the manipulator dynamics in (2), the collision avoidance constraints

$$B_l(d_j(t), x(t)) \ge 0, \; \forall t \in [t_i, t_i + h], \; \forall j \in \mathcal{N}_o(t) \quad (6)$$

where $l = 1, 2, \ldots, n$. Then, the performance index in (5) may be chosen as

$$J(u(t)) \triangleq \frac{1}{2}\int_{t_i}^{t_i+h} \Big\{w_e\|E(t)\|^2 + w_u\|u(t)\|^2\Big\}dt + \frac{1}{2}w_f\|E(t_{i+1})\|^2, \quad (7)$$

where $E(t)$ represents the tracking error in the Cartesian coordinates (or in the joint variable space), and w_e, w_u, and w_f are non-negative weighting factors for the corresponding variables.

Let $h_c \triangleq t_{i+1} - t_i$ be the sampling interval of our receding horizon scheme. The receding horizon collision-free trajectory generation scheme based on the FHOCP in (5) is described by the following algorithm.

Receding Horizon Collision-Free Trajectory Planning
Data: x_0 of the system (2), $G_d(t)$ for $t \in [0, T]$, $t_0 = 0$
Solve the FHOCP (5) for $t \in [0, t_1]$ with $\mathcal{N}_o(0)$ and compute the optimal trajectory $x^*(t)$ and $G_d^*(t)$ for $t \in [0, t_1]$
Set $i = 0$.
loop
  if $t = t_i$ **then**
    Apply $G_d^*(t)$ for $t \in [t_i, t_{i+1}]$.
    if $t_{i+1} \ge T$ **then**
      Terminate
    end if
    Update $\mathcal{N}_o(t_i)$
    Solve the FHOCP (5) with initial state $x^*(t_{i+1})$ and $\mathcal{N}_o(t_i)$
    Compute $x^*(t)$ and $G_d^*(t)$ for $t \in [t_{i+1}, t_{i+2}]$
    $i = i + 1$
  end if
end loop

Figure 1 shows the graphical representation of the above algorithm.

There are several important steps in implementing the above algorithm in practice and we employ the two-link manipulator to elaborate them in the next section.

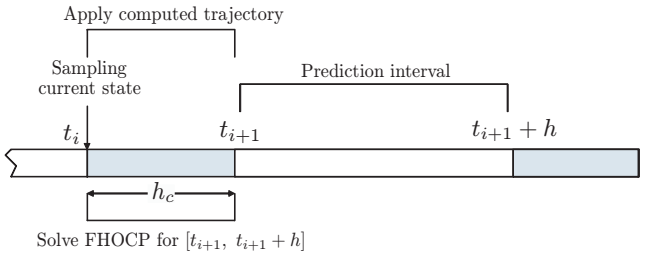


Fig. 1. Graphical illustration of the RH trajectory generation algorithm

III. Two-link Manipulator Example

A. Application to a Two-link Manipulator

Consider a planar two-link manipulator, i.e. $n = 2$, as shown in Fig. 2. In this case, the manipulator dynamics in (1) becomes (Dependencies on t is omitted in the remaining for simplicity.)

$$\begin{aligned} &M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau \\ &M(q) = \begin{bmatrix} \alpha + 2\beta\cos q_2 & \delta + \beta\cos q_2 \\ \delta + \beta\cos q_2 & \delta \end{bmatrix}, \\ &C(q,\dot{q}) = \beta\sin q_2 \begin{bmatrix} -\dot{q}_2 & -(\dot{q}_1+\dot{q}_2) \\ \dot{q}_1 & 0 \end{bmatrix}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} \alpha &= J_1 + m_1 r_1^2 + J_2 + m_2(\ell_1^2 + r_2^2) \\ \beta &= m_2 \ell_1 r_2, \quad \delta = J_2 + m_2 r_2^2 \end{aligned}$$

with m_l, J_l, and ℓ_l denoting the mass, the mass moment of inertia, and the length of the l-th link, respectively. r_l is the distance to the center of mass of each link, where J_l is evaluated. The exemplary values for the mechanical parameters to be used in the remaining are listed in Table I.

Link 1 parameter	unit	Link 2 parameter	unit
$\ell_1 = 0.32$	m	$\ell_2 = 0.21$	m
$r_1 = 0.16$	m	$r_2 = 0.046$	m
$m_1 = 9.244$	kg	$m_2 = 3.529$	kg
$J_1 = 0.2097$	kg · m^2	$J_2 = 0.0206$	kg · m^2

TABLE I
Mechanical Parameters

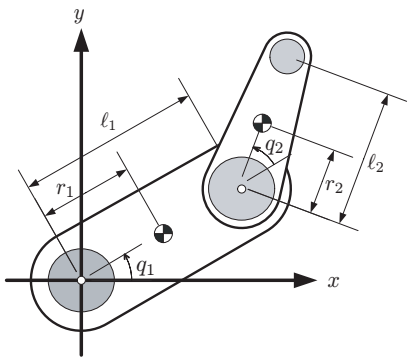


Fig. 2. Configuration of a two-link robot arm

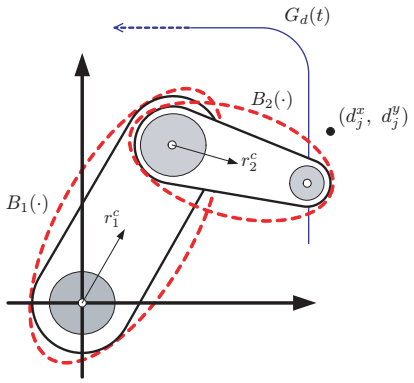


Fig. 3. The ellipses for collision avoidance ($B_l(p, q(t)) = 0$), the desired trajectory $G_d(t)$ and the obstacle point d_j

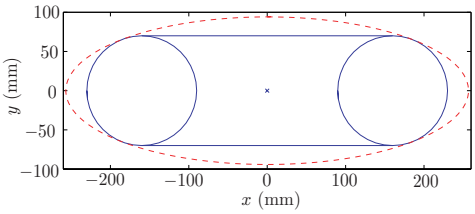
Now, we define $B_l(p, q(t)) = 0$ as the smallest ellipse containing the $l-$th link, whose center is fixed on somewhere in the $l-$th link as shown in Fig. 3. More specifically,

$$\begin{aligned} B_l(d_j, q(t)) = &\frac{[(d_j^x - c_k^x)\cos\phi_l + (d_j^y - c_k^y)\sin\phi_l]^2}{a_l^2} \\ &+ \frac{[(d_j^y - c_l^y)\cos\phi_l - (d_j^x - c_l^x)\sin\phi_l]^2}{b_l^2} - 1, \end{aligned} \tag{9}$$

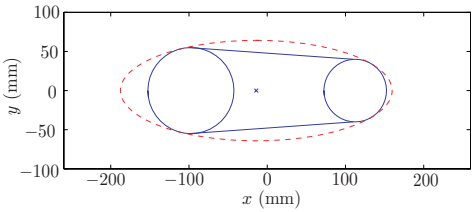
where $\phi_l = \sum_{i=1}^{l} q_i$, $l = 1, 2$, and

$$c_1 = r_1^e \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix}, \ c_2 = \ell_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix} + r_2^e \begin{bmatrix} \cos(q_1+q_2) \\ \sin(q_1+q_2) \end{bmatrix}. \tag{10}$$

r_l^e denotes the distance from the center of the $l-$th joint to the center of the ellipse $B_l(\cdot)$, as shown in Fig. 3. Equation (9) is obtained by rotating the minimal ellipse at the origin by ϕ_l and translating it on the $l-$th link. Figure 4 shows $B_1(\cdot)$ and $B_2(\cdot)$ for our two-link manipulator example. Parameters of ellipses are obtained by the method in [13] and are listed in Table II.



(a) $B_1(\cdot)$



(b) $B_2(\cdot)$

Fig. 4. Minimum area ellipses for the links. 'x' denotes the center of the ellipse.

Link 1	Link 2
$a_1^2 = 6.5957E-02$	$a_2^2 = 3.0155E-02$
$b_1^2 = 8.8631E-03$	$b_2^2 = 4.1179E-03$
$r_1^e = 1.6000E-01$	$r_2^e = 8.3545E-01$

TABLE II

PARAMETER VALUES OF THE MINIMUM AREA ELLIPSES

For the tracking error term $E(t)$ in (7) we use the tracking error in the Cartesian coordinates as follows.

$$E(t) = G_d(t) - \mathcal{K}(q(t)). \tag{11}$$

Comparing with the tracking error in the joint variable space, $E_q(t) = \mathcal{K}^{-1}(G_d(t)) - q(t)$, the above formulation only uses the forward kinematics $\mathcal{K}(\cdot)$, and therefore is immune from any singularity that may be induced by the inverse kinematics.

In order to state the optimal trajectory problem as an end-point problem defined on $s \in [0,1]$, we rescale the state dynamics of the manipulator using the prediction interval h, $t = t_i + hs$, and augment the physical state with an additional component, x^5,

$$x^5(s) \triangleq \frac{1}{2}\int_0^s \Big\{ w_e \|E_i(\tau)\|^2 + w_u \|u(s)\|^2 \Big\} d\tau. \tag{12}$$

Note that we abuse notations $E(\cdot)$, $x_i(\cdot)$, etc. for simplification[2]. The resulting dynamics of the manipulator have the form of

$$\frac{dx(s)}{ds} = h \begin{bmatrix} \dot{q}(s) \\ M^{-1}(q(s))\{u(s) - C(q(s)) - g(q(s))\} \\ \frac{1}{2}\Big\{ w_e \|E(s)\|^2 + w_u \|u(s)\|^2 \Big\} \end{bmatrix} \tag{13}$$

with the initial state $x(0)$ given. We will denote the solution of the above dynamic equation by $x(s,u)$, with $s \in [0,1]$.

The optimal control problem we need to solve is of the form

$$\min_{u \in \mathcal{U}} \Big\{ x^5(1,u) + \frac{1}{2}\|E(1)\|^2 \Big\} \tag{14}$$

subject to

$$B_l(d_j(s), x(s)) \geq 0 \tag{15}$$

for all $s \in [0,1],\ l = 1,2$ and $j \in \mathcal{N}_o(t)$, and the dynamics (13).

Since the above optimization problem is infinite-dimensional, we need to discretize the dynamics and convert the problem into a finite-dimensional one. In this paper, we follow the treatment in [14] and use Euler's method to obtain

$$\bar{x}(s_{k+1}) - \bar{x}(s_k) = \Delta f(\bar{x}(s_k), \bar{u}(s_k)),\ \bar{x}(0) = x(0), \tag{16}$$

with $\Delta \triangleq 1/N$, $N \in \mathbb{N}$ (i.e. integer), $s_k \triangleq k\Delta$ and $k \in \{0,1,\ldots,N\}$. We use an over-bar to distinguish between the exact variables and the discretized variables. We will denote the solution of the discretized dynamics by $\bar{x}(s_k, \bar{u}) \in \mathcal{X} \subset \mathbb{R}^5$, $k = 0,1,\ldots,N$, with

$$\bar{u} \triangleq (\bar{u}(s_0), \bar{u}(s_1), \ldots, \bar{u}(s_{N-1})). \tag{17}$$

[2]One may introduce $E_i'(s) \triangleq E(t_i + hs)$ and so on for more rigorous notations.

Finally, we obtain the following discrete-time optimal control problem:

$$\min_{\bar{u} \in \bar{\mathcal{U}}} \Big\{ \bar{x}^5(1,\bar{u}) + \frac{1}{2}\|\bar{E}(1)\|^2 \Big\} \tag{18}$$

subject to the discretized dynamics (16), and the discretized collision avoidance constraints:

$$B_l(d_j, \bar{q}(s_k)) \geq 0,\ \forall k = 1,2,\ldots,N,\ l = 1,2, j \in \mathcal{N}_o. \tag{19}$$

B. Reduction of the Computational Load using the External Active-set Strategy

The discrete-time optimal control problem we derived above has many constraints due to the discretization. For example, from (19), there are $N_o \times N \times n = 2NN_o$ collision avoidance inequality constraints. Solving a discrete-time optimal control problem with a large number of nonlinear inequality constraints is computationally very expensive, and it may make the proposed RHC-based collision avoidance algorithm impractical. However, since potential collisions are confined to relatively short segments of the manipulator trajectories and many obstacle points are not reachable within the prediction horizon h, most of the collision avoidance inequalities are inactive. Moreover, in the two-link manipulator example, most of avoidable obstacles make conflicts with the second link, and the collision avoidance constraints for the first link are redundant in most cases. If those inactive inequality constraints are cleverly excluded, then the computation time required to solve the discrete-time optimal control problem would be reduced and faster manipulator motion would be possible using the proposed RHC-based collision-free planning algorithm. One may suggest to remove those unreachable constraints by computing reachable space. In this case, however, it is impossible to prove the convergence of the optimization algorithm to a solution of the original problem with the full set of constraints. Recently, Chung et. al. [1] proposed an external active-set strategy, which can exclude inactive inequality constraints from the computation while maintaining the convergence property of *any* optimization algorithm.

In each iteration, the algorithm computes a set of ϵ-active inequality constraints, merges it to the existing active-set, and submits it to the optimization solver for a fixed number of iteration. In the case of our two-link manipulator example, the set of ϵ-active collision avoidance constraints is

$$\mathbf{q}_\epsilon(u) \triangleq \{(l,j,k) \in \mathbf{q} \mid -B_l(d_j, \bar{q}(s_k)) \leq \psi_+(u) - \epsilon\}, \tag{20}$$

where

$$\psi_+(u) \triangleq \max\{0, \psi(u)\}, \tag{21}$$

and

$$\psi(u) \triangleq \max_{(l,j,k) \in \mathbf{q}} [-B_l(d_j, \bar{q}(s_k))]. \tag{22}$$

$\mathbf{q}$ is a set of all possible triples (l,j,k) for all $l = 1,2$, $j = 1,2,\ldots,N_o(s)$, and $k = 1,2,\ldots,N$. Then $\mathbf{q}_\epsilon$ is merged with the existing active-set, i.e. $\mathbf{q}_{i+1} = \mathbf{q}_i \cup \mathbf{q}_\epsilon$, and the problem submitted to the optimization solver for N_ϵ

iterations. This procedure is repeated until the optimal solver returns with a stationary solution, and it is feasible for the entire constraint set $\mathbf{q}$.

Since an inequality constraint $B_l(d_j, \bar{q}(s_k)) \geq 0$ cannot be included in $\mathbf{q}_\epsilon$ unless the minimal distance between the $j-$th obstacle point and the $l-$th ellipse is less than or equal to $\epsilon \geq 0$, the collision avoidance constraints corresponding to unreachable obstacle points are automatically excluded, and hence the computational load is reduced by solving the discretized FHOCP problem with smaller number of inequalities.

In each active-set iteration, ϵ is determined by following adaptation:

$$\epsilon \triangleq \min\{\psi_+(u_i), \epsilon_{max}\}. \tag{23}$$

IV. Numerical Results

The proposed receding horizon trajectory generation scheme with the external active-set strategy was implemented in MATLAB. The numerical experiments were performed using TOMLAB V5.7 [15] with MATLAB V7.8 running in Windows XP on a desktop computer equipped with the Intel i7 CPU 920 at 2.67GHz processor and 3GB of RAM. SNOPT 6.2 [16] was used as the main optimization solver.

Figure 5 shows the initial configuration of the robot, $x_0 = [\pi/3, \ -\pi/6, \ 0, \ 0, \ 0]^T$, and the obstacle points. The simulation begins with four obstacle points, specified as Set 1 in Figure 5. These obstacle points are located to cause collisions with the manipulator whose end effector is commanded to track the reference trajectory given by

$$G_d(t) = \begin{bmatrix} A_d \\ B_d \cos w_d t \end{bmatrix}, \quad \begin{bmatrix} A_d \\ B_d \end{bmatrix} = \mathcal{K}\left(\begin{bmatrix} x_0^1 \\ x_0^2 \end{bmatrix} \right) \tag{24}$$

with $w_d = 0.02\pi$.

We assume that a new set of obstacles, labeled as Set 2 in Figure 5, is detected at $t = 25$, which also should be avoided. Therefore $N_o(t) = 4$ for $0 \leq t < 25$, and $N_o(t) = 8$ for $25 \leq t$. Table III summarizes the parameter values used in the simulation.

Figure 6 shows the stick representation of manipulator motion with the bounding ellipses, which illustrates that the algorithm successfully generates the collision-free trajectory. As shown in Figure 7, the obstacles cause the manipulator end effector to move away from the reference trajectory. Meanwhile, the FHOCP made reasonable trade-off between the trajectory tracking and the collision avoidance. The generated trajectory in the Cartesian space is also shown in Figure 5.

In the simulation, the number of collision avoidance constraints is quite large. There are 512 collision avoidance

Parameter	Value	Parameter	Value
N	64	h	10 (sec)
W_u	1	W_e	2
h_c	5 (sec)	W_f	2
ϵ_{max}	0.1	N_ϵ	50

TABLE III
Simulation Parameters

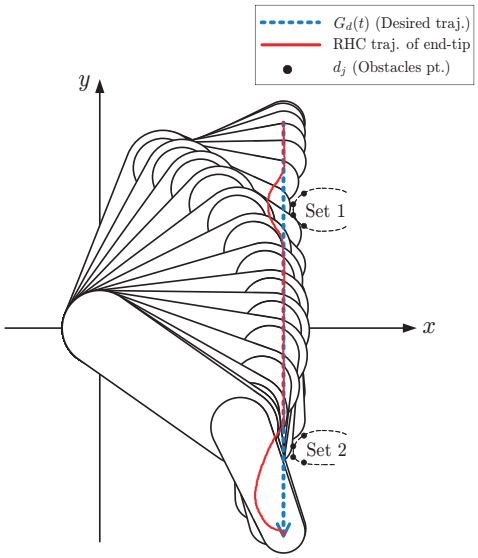


Fig. 5. Simulation scenario with the fixed obstacle points (Set 1) and the spontaneous obstacle points (Set 2) that show up at $t = 25$ s. The red solid line shows the actual end-tip trajectory generated by the proposed algorithm.

constraints for $0 \leq t < 25$, and 1028 for $25 \leq t$. If we run the FHOCP problem with the full set of collision avoidance constraints for obstacle points in Set 1, i.e., without using the external active-set strategy, it takes normally a few thousand seconds to compute the optimal solution using the same computer. In contrast, the external active-set strategy drastically reduced the computation time below 60 seconds as shown in Fig. 8(a). This is because only a fraction of constraints (less than 100) were included in the computation as shown in Fig. 8(b). Note that, in spite of such a significant reduction in computation time, the proposed algorithm still needs to be improved for the practical application because the trajectory of our choice is assumed to be updated every 5 seconds. Clearly, we will need to use solvers other than SNOPT and implement the code set in a dedicated realtime machine independent from MATLAB. Currently, several options are under our investigation to further reduce the computation time and are explained in more detail in the Future Work.

V. Conclusions and Future Work

In this paper, we investigated the feasibility of the collision-free trajectory generation using a receding horizon strategy for robotic manipulators. The distinguished features of the proposed method are 1) on-line optimal trajectory generation under dynamic environment, 2) imposition of hard constraints on state variables and control inputs, 3) being immune to manipulator singularity issues, 4) drastic reduction in the computation time through the external active-set strategy to exclude inactive constraints effectively.

To apply our algorithm to real-world applications, the computation time must be further reduced. In solving numerical optimal control problems, it is well known that one can save a significant amount of computation time by using the adaptive discretization, i.e. coarse discretizations at the start of the computation and then refining it as the solution

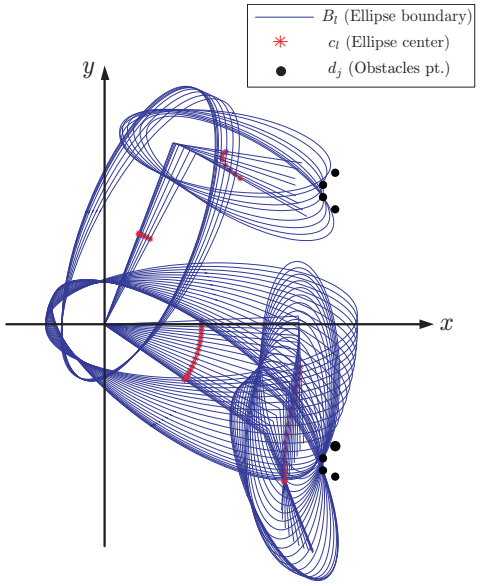


Fig. 6. Traces of enclosing ellipses around obstacle points. The marks '*' represent the center of ellipses on the links, and the marks • the obstacle points, respectively.

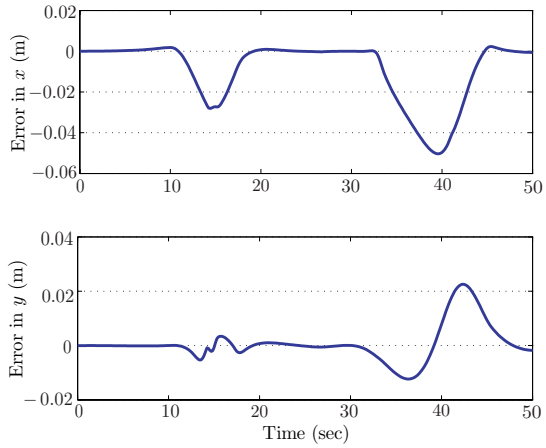
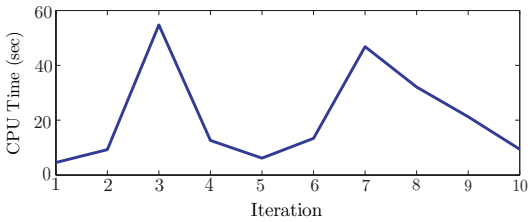
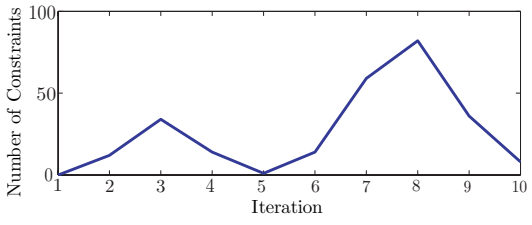


Fig. 7. Tracking error in the Cartesian space



(a) CPU time



(b) Number of constraints used

Fig. 8. Computation saving by the external active-set strategy

is approached. A set of adaptive discretization rules can be found in Section 3.3.3 of [14]. When the parameters for these rules are well chosen, it is typical to achieve a tenfold reduction in computing time, compared to using a fine discretization throughout the entire computation. More appropriate for the setting described in this paper may be to utilize the optimal discretization scheme described in [17], which does not depend on fine tuning a set of parameters. The optimal discretization scheme sets up an optimization problem whose solution consists of the number of discretization stages to use and the number of discretizations in each stage as well as the number of iterations to perform in each stage, so as to minimize the computing time necessary to reduce the initial error by a pre-assigned factor. This scheme indeed depends on problem parameters, but they can be easily estimated as explained in [17]. In the short term, we will test our algorithm on a more powerful computer with various algorithms including the state-of-the-art interior point method [18].

REFERENCES

[1] H. Chung, E. Polak, and S. S. Sastry, "On the Use of Outer Approximations as An External Active Set Strategy," Journal of Optimization Theory and Applications, vol. 146, no. 1, pp. 51-75, 2010.

[2] R. Johansson, "Quadratic optimization of motion coordination and control," *IEEE TAC*, vol. 35, no. 11, pp. 1197-1208, 1990.

[3] B. S. Chen, T.-S. Lee, and J.-H. Feng, "A nonlinear $\mathcal{H}_\infty$ control design in robotics systems under parametric perturbation and external disturbance," *Int. J. of Control*, vol. 59, no. 12, pp. 439-461, 1994.

[4] J. park and W. K. Chung, "Analytic nonlinear $\mathcal{H}_\infty$ inverse-optimal control for euler.lagrange system," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 847-854, 2000.

[5] J. E. Bobrow, F. C. Park, and A. Sideris, "Recent advances on the algorithmic optimization of robot motion," *Lecture Notes in Control and Information Sciences*, vol. 340, pp. 21-41, 2006.

[6] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733-764, 2003.

[7] D. H. Shim, H. Chung, and S. Sastry, "Conflict-free navigation in unknown urban environments," *IEEE Robotics and Automation Society Magazine*, vol. 13, pp. 27-33, 2006.

[8] T. Fan and C. W. de Silva, "Dynamic modelling and model predictive control of flexible-link manipulators," *International Journal of Robotics and Automation*, vol. 23, no. 4, pp. 227-234, 2008.

[9] P. Poignet and M. Gautier, "Nonlinear model predictive control of a robot manipulator," in *International Workshop on Advanced Motion Control*, 2000, pp. 401-406, nagoya, Japan.

[10] B. Song and A. J. Koivo, "Nonlinear predictive control with application to manipulator with flexible forearm," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 5, pp. 923-932, 1999.

[11] Y. Kawai, Y. Nakaso, S. Mimoto, and M. Fujita, "Experiments on stabilizing RHC of a direct drive manipulator," *Electronics and Communications in Japan*, vol. 91, no. 5, pp. 748-754, 2008.

[12] J. J. Craig, *Introduction to Robotics*. Addison-Wesley, 1989.

[13] N. Moshtagh, "Minimum volume enclosing ellipsoids," unpublished note. http://www.mathworks.com/matlabcentral/fileexchange/9542

[14] E. Polak, *Optimization: Algorithms and Consistent Approximations*, ser. Applied Mathematical Sciences, Springer, 1997, vol. 124.

[15] K. Holmström, A. O. Göran, and M. M. Edvall, *User's Guide for TOMLAB*, Tomlab Optimization Inc., December 2006.

[16] W. Murray, P. E. Gill, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, pp. 979-1006, 2002.

[17] L. He and E. Polak, "Effective diagonalization strategies for the solution of a class of optimal design problems," *IEEE Transactions on Automatic Control*, vol. 35, no. 3, pp. 258-267, 1990.

[18] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Math. Programming*, vol. 106, no. 1, pp. 25-57, 2006.