

LP BASED GLOBAL MINIMIZATION OF PEAK BASE REACTION FORCE OF MANEUVERING ROBOTS

Matthias Schmid*

Department of Mechanical & Aerospace Engineering
University at Buffalo, The State University of New York
Amherst, NY 14260-4400

Lukas Ramrath†

Department of Mechanical & Aerospace Engineering
University at Buffalo, The State University of New York
Amherst, NY 14260-4400

T. Singh‡

Department of Mechanical & Aerospace Engineering
University at Buffalo, The State University of New York
Amherst, NY 14260-4400

ABSTRACT

A method for computing the optimal control to minimize a robot's peak base reaction force, while avoiding obstacles, is presented. It contains the manipulator dynamics, initial and final conditions and obstacle constraints. Using the assumption that minimal peak base reaction force control is equal to time-optimal control, an iterative approach is used to find the path and control that globally minimizes the robots peak base reaction force. The conditions for optimality are derived and incorporated into an algorithm which uses linear programming to solve the given problem. Equations describing the system dynamics and the obstacle position are mapped into the center of mass space, a convenient space for path planning. The method is demonstrated for a general maneuver in the center of mass space and for a two-link manipulator which includes obstacle constraints.

INTRODUCTION

The reduction of the peak base reaction force during robot maneuvers is important for many kinds of application. These applications might be of industrial, military or research nature. Optimal motions can improve the life-cycle of the robot and its environment, and cost effectiveness. Additionally, minimization of the peak base reaction force reduces the impact of the robot on its environment. In a spacecraft, for example, base reaction forces are often undesired effects which can be seen as a disturbance to the general

spacecraft behavior (e.g. attitude) initiating undesired movements and oscillations. The base reaction forces of a manipulator, attached to a mobile base, can cause overturning of the base.

To date, several approaches to minimize these kind of disturbances exist. If satellite manipulators are considered, most attention was paid to minimize the attitude disturbances caused by base reaction torques. Some techniques were developed which are able to minimize the disturbance while driving the satellite back to a prescribed position (Schulz et. al^{3,4}). Some suggested a partitioned approach to the control of the manipulator and the satellite. Other approaches are based on robot link motions that dynamically cancel. For a limited area in the workspace, zero base reaction force can be achieved (Papadopoulos¹¹). Only few approaches to the global optimization of a manipulators peak base reaction force exist. Kazerounian and Wang used the relationship between local acceleration optimization and global velocity optimization. Doggett et. al^{2,5,6} proposed a computationally efficient technique which leads to a path having a peak force within 5% of the optimal path. This approach is based on the parameterized equations of the C^2 smooth path that globally minimizes the Euclidean norm of a robot's peak base reaction force. Additionally, Doggett et. al introduce the CM space (center of mass space), a convenient space for planning and calculating robot motions.

This paper develops an approach to find the global optimal path which minimizes the robot's peak base reaction force while avoiding obstacles in the 2-D Cartesian space. The transformation from Cartesian to CM space is used, as a multi-body robot can be transformed into a point mass simplifying path cal-

*Graduate Student. Email: mjschmid@eng.buffalo.edu

†Graduate Student. Email: lramrath@eng.buffalo.edu

‡Associate Professor, Senior Member AIAA. Email: tsingh@eng.buffalo.edu

culations and path planning. The CM space is a Cartesian-like space and allows direct application of control techniques. Problems of mapping into the CM space are addressed.

The method is based on an iterative approach to compute the peak base reaction force minimizing path. It uses the technique of linear programming (LP) while incorporating the robots dynamic constraints, obstacle constraints and initial and final conditions.

Basic assumptions are the absence of gravity and the constancy of the robot mass, obstacle position, and height during the maneuver.

The conditions for a path which globally minimizes the peak base reaction force are derived. It is shown that the time-optimal path is equivalent to the path that minimizes the peak base reaction force. Techniques which find the time-optimal path are applied to the problem and based on these conditions of optimality, the general solution of the control is presented. Furthermore, the iterative approach to the problem is described. It is based on a LP technique which has the advantage of being computationally efficient. In order to be able to apply LP, constraints have to be linearized.

Two numerical example problems will be solved.

As this method operates iteratively, it assures that the global optimal path is found.

DEVELOPMENT

ASSUMPTIONS

Several assumptions have to be made for the iterative approach:

- The robot mass and the obstacle position are constant during the whole maneuver.
- The initial and final configuration is given in terms of \vec{x}_0 and \vec{x}_{T_f} .
- The problem is transferred from a desired maneuver in the Cartesian space to a maneuver in the CM space.

The problem is then formulated by moving a point mass with a minimal peak input over an given obstacle (see figure 1).

COST FUNCTION

The peak base reaction force depends on the torque at the manipulator's links. Newton's first law of movement states that the torques, and therefore the peak base reaction force, are proportional to the acceleration of the center of mass of the robot ($\vec{\tau} = l_c \vec{F}_c$, $\vec{F}_c = m \vec{a}_c$). Therefore, we are looking for the optimal control input $[a_{cx} \ a_{cy}]$ which can be transformed into joint angles. This is the main reason for using the CM space.

If there would be no restriction on the maneuver time T_f , the optimal solution would be $\vec{a} = \vec{u} \rightarrow \vec{0}$.

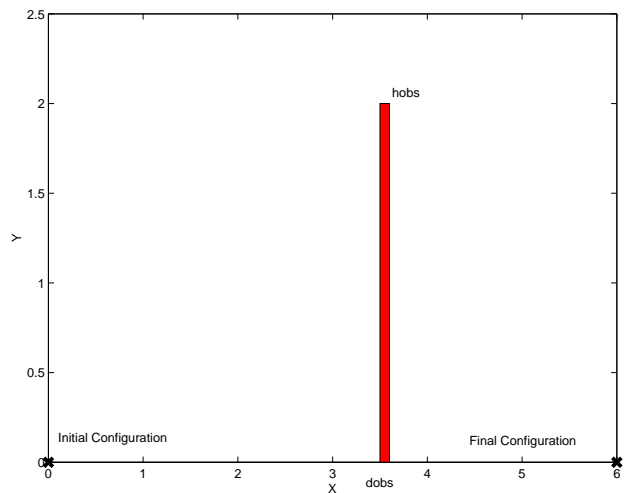


Fig. 1 Moving objective

Thus, the final time T_f has to be given by the user and/or the problem.

Furthermore, the control vector is restricted since we want to minimize the acceleration. Consider the following iterative approach:

Given any upper and lower bound on the control, the time optimal solution is found. If the final time achieved by the time optimal solution T_{to} is less than the problem given time T_f , the restrictions on the control can be decreased until the time optimal solution exactly meets the given final time. Then, there exists no other control with stricter boundaries on the control input which fulfills the final time constraint T_f .

Therefore, the time optimal profile with a given final time, where the bounds on the control vector are the unknown parameters, establishes the cost function.

TIME-OPTIMAL PATH

As the time optimal path and the path of minimal peak base reaction force are equivalent, the problem to drive the robot from an initial to a final position in the CM space can be formulated as the following optimization problem:

$$J(\vec{u}(t)) = \int_{T_0}^{T_f} 1 \, dt = T \quad (1)$$

subject to

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}$$

$$\sqrt{(u_x^2 + u_y^2)} \leq M,$$

the initial conditions

$$\vec{x}(0) = \vec{x}_0,$$

and the final conditions

$$\vec{x}(T_f) = \vec{x}_{T_f}$$

where the state vector is defined as

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix}$$

As the system is described in the CM Space, A and B become

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Obstacle constraints are

$$x_1(T_{obs}) = d_{obs}$$

and

$$x_3(T_{obs}) = h_{obs}$$

where d_{obs} is the distance to the obstacle and h_{obs} is the height of the obstacle (see figure 1).

The Hamiltonian of equation (1) is defined as

$$\begin{aligned} \mathcal{H} &= 1 + \lambda_1 x_1 + \lambda_2 u_1 + \lambda_3 x_2 + \lambda_4 u_2 \\ &= 1 + \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} \begin{bmatrix} \lambda_1 & \lambda_3 \end{bmatrix} + \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \begin{bmatrix} \lambda_2 & \lambda_4 \end{bmatrix} \end{aligned}$$

The optimality conditions are:

$$\frac{\partial \mathcal{H}}{\partial \vec{x}} = -\dot{\vec{\lambda}}$$

$$\frac{\partial \mathcal{H}}{\partial \vec{\lambda}} = \dot{\vec{x}}$$

The third condition is achieved by applying Pontryagin's minimum principle

$$\vec{u}(t) = -M \frac{\vec{\lambda}(t)B(t)}{\|\vec{\lambda}(t)B(t)\|} = -M \frac{\begin{bmatrix} \lambda_2 & \lambda_4 \end{bmatrix}}{\|\begin{bmatrix} \lambda_2 & \lambda_4 \end{bmatrix}\|}$$

defining the optimal control vector for the given problem. \vec{u} points in the negative direction of the vector $\begin{bmatrix} \lambda_2 & \lambda_4 \end{bmatrix}$ with the length M . Therefore, the control input is a vector lying on a circle. Pontryagin's minimum principle can be used as the control input is always restricted. The goal is to find the minimal restriction that allows a feasible solution.

CONSTRAINTS FOR OBSTACLE AVOIDANCE

The constraints for obstacle avoidance were stated while formulating the time-optimal problem. The first constraint is that at the time the point mass reaches the obstacle, the constraint $x_{1obs} = d_{obs}$ must be met. The necessary condition for avoiding the obstacle in y direction is $x_{3obs} \geq h_{obs}$. For a time optimal movement without an obstacle, the optimal path would be the direct connection between initial and final configuration (i.e. no movement in y direction). If an obstacle is present, optimality is given by the smallest displacement from the time optimal path without an obstacle. Thus, $x_{3obs} = h_{obs}$ becomes the obstacle constraint in y direction as this means a minimal perturbation from the optimal path without an obstacle.

PROBLEMS

Hamiltonian based approaches have to deal with the existence of a large number of local minima. The search for the global optimum therefore becomes time consuming and computationally expensive.

SOLUTION TECHNIQUE

To solve the given problem without running into the problems presented, an iterative approach is used. This guarantees that every feasible solution which minimizes the peak base reaction force is found.

Two varying parameters are defined:

1. T_{obs}

As x_{1obs} is given, the time to the obstacle T_{obs} depends on the magnitude of x acceleration (u_1). T_{obs} only depends on the motion in x direction. As the acceleration u_1 is not known, every T_{obs} in the interval $[T_0, T_f]$ has to be checked.

2. M

The main purpose of the algorithm is to minimize the magnitude of the control input M . The minimal magnitude for a given set of parameters (including initial and final conditions, obstacle constraints and T_{obs}) is found as the limit to a set of parameters that still yield a feasible solution. Combined with the parameter T_{obs} , the minimal M has to be found for every possible T_{obs} .

Both parameters set up an algorithm consisting of two loops. In an outer loop, T_{obs} is varied from $[T_0, T_f]$. An inner loop calculates the minimal M for every given T_{obs} which still yields a feasible solution.

The output of the algorithm is a distribution of minimal M versus every possible T_{obs} . In this distribution, the *global* minimum M can be found.

DESCRIPTION

The iterative approach using LP can be seen in figure 2. First, the interval from $[T_0, T_f]$ is divided into i intervals which have the length Δt . i can be chosen

by the user (increasing i corresponds to greater computational needs). Then T_{obs} is defined and passed to an inner loop. The inner loop varies the magnitude of the control. Using the bisection method, the smallest magnitude, for which a feasible solution that meets all constraints, is found by formulating and solving a LP problem. The Bisection method works as follows:

1. If the initial M_{max} is not feasible, it is increased until a feasible solution is found.
2. Check if the difference between upper bound M_{right} and lower bound M_{left} is smaller than a certain convergence criteria. If so, this M is considered the minimal M for this T_{obs} for which the control yields a feasible solution. If not, the algorithm switches to step 3.
3. The middle of the upper bound M_{right} and the lower bound M_{left} becomes M_m .
4. The feasibility of M_m is checked. If it is feasible the upper bound is set to M_m . If it is not feasible, the lower bound becomes M_m .
5. Go back to step 2

Once the minimal control input for a certain T_{obs} is found, T_{obs} is increased by Δt . If the final time is not reached and the bisection algorithm is called again.

THE LP PROBLEM

As linear programming is used in order to solve for a feasible solution, the problem and the constraints have to be expressed in linear form.

- System dynamics
As the constraint due to the system dynamics are given in form of first order differential equations, the state space representation is discretized. The time interval from $[T_0, T_f]$ is divided into m steps. This yields the following discrete representation:

$$\vec{x}(k+1) = \Phi\vec{x}(k) + \Gamma\vec{u}(k)$$

By discretizing, $4m$ linear constraints are created as each time step represents one linear constraint for all four states.

- Initial, final states
The initial and final states are part of the discretization of the state space representation and therefore incorporated into the linear constraints created above.
- Obstacle constraints
The optimality condition demands that $x_{1obs} = d_{obs}$ and $x_{3obs} = h_{obs}$. These constraints are implemented by formulating a subsystem where the final conditions are the boundary conditions of the obstacle (initial conditions are the conditions

of the original problem). In other words, the same discretized system dynamics constraints are used in a new interval $[T_0, T_{obs}]$. Note that only two states are constraint, the other two are free. This creates $2k_{obs}$ linear constraints where k_{obs} is the number of time steps to the obstacle which varies for every T_{obs} .

- Input constraint (Circle constraint)
The original constraint that the the control vector has to lie on a circle is not usable for LP. Therefore, the circle is approximated by a user defined number of lines. The circle is subdivided into $4n$ equally sized segments where n is the number of segments in one quadrant chosen by the user. The points on the circle defined by these angles are connected and create another set of linear constraints. As these constraints have to be met for each time step, this yields $m \cdot 4n$ additional constraints. Figure 3 shows the approximation for one quadrant, 5 segments per quadrant and a maximum magnitude of $M = 1$.

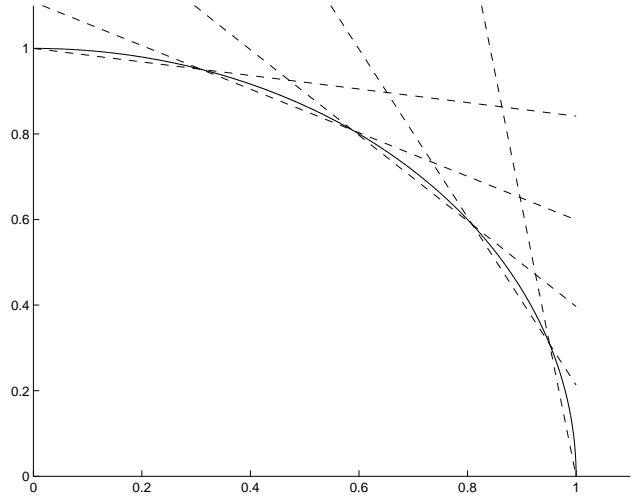


Fig. 3 Circle approximation

The linear programming problem then becomes:

$$\min f^T \vec{u}$$

subject to

$$\begin{aligned} \vec{x}(k+1) &= \Phi\vec{x}(k) + \Gamma\vec{u}(k) \\ \vec{x}(0) &= \vec{x}_0 \\ \vec{x}(k_f) &= \vec{x}_f \\ x_1(k_{obs}) &= d_{obs} \\ x_3(k_{obs}) &= h_{obs} \\ u_2(k) &\leq m_l u_1(k) + b_l \\ u_2(k) &\leq -m_l u_1(k) + b_l \\ u_2(k) &\leq m_l u_1(k) - b_l \\ u_2(k) &\leq -m_l u_1(k) - b_l \\ l &= 1, n \end{aligned}$$

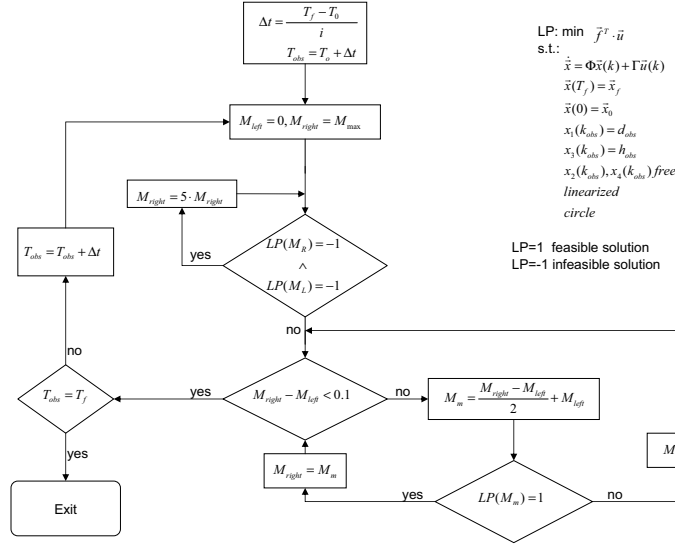


Fig. 2 Flowchart

where m_l the slope of the l^{th} linear approximation curve and b_l its intersection with the y -axis. f^T is defined as a vector of 0's. Note that f^T has no special meaning as the linear programming only checks the feasibility of a solution.

NUMERICAL EXAMPLES

In order to show the advantages of the presented linear approach, the method will be tested on two problems:

1. General problem formulation

To analyze certain characteristics of the linear approach, a general problem is chosen on which analysis is performed. The goal is to drive the center of mass from an initial configuration of

$$\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ to a final position of } \vec{x}_{T_f} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

in a time of 1sec. The obstacle is located at $d_{obs} = 0.5$ and its height is $h_{obs} = 0.5$. The dependence of the results on the number of linear circle approximation curves n will be examined. n is varied over an interval $[0, 5]$. The interval $[T_0, T_f]$ is divided into 18 time steps Δt .

2. Two-link manipulator

The problem posed is to drive the manipulator from an initial position to a final position while avoiding an obstacle. The final condition is given by the tip touching a box on a table where the table itself presents the obstacle. The two-link manipulator has the following properties: Length of

the links $l_{1,2} = 4m$, mass of the links $m_{1,2} = 0.5kg$ and the mass centers concentrated at $l_{c1,c2} = 2m$ (see figure 4). Results visualize the relationship

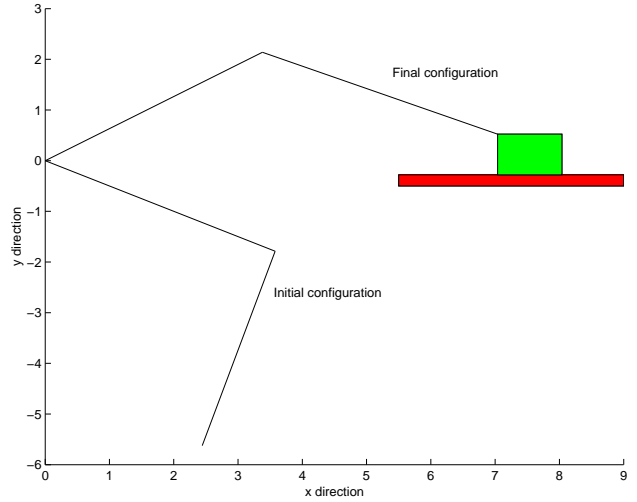


Fig. 4 Test-problem

between T_{obs} and the minimal M , the trajectory in the CM space and the trajectory in the Cartesian space. In this case the interval $[T_0, T_f]$ is divided into 14 time steps Δt . The time in which the maneuver should be performed is chosen as $T_f = 5$ sec, the number of circle constraints in one quadrant is $n = 5$.

RESULTS

GENERAL PROBLEM

Figures 5 and 6 show results depending on the variation of the number of linear circle approximations n .

Figure 5 presents the control input for respectively $n = 1, 3, 5$. Figure 6 presents the minimal input magnitude M , T_{obs} and the squared difference between the m^{th} and the $(m^{th} - 1)$ control input where $m \in [2, 5]$. All results are summarized in table 1 where $\Delta \vec{u}_x = \sum_{k=1}^{200} (\vec{u}_{x,m} - \vec{u}_{x,m-1})^2$ and $\Delta \vec{u}_y = \sum_{k=1}^{200} (\vec{u}_{y,m} - \vec{u}_{y,m-1})^2$.

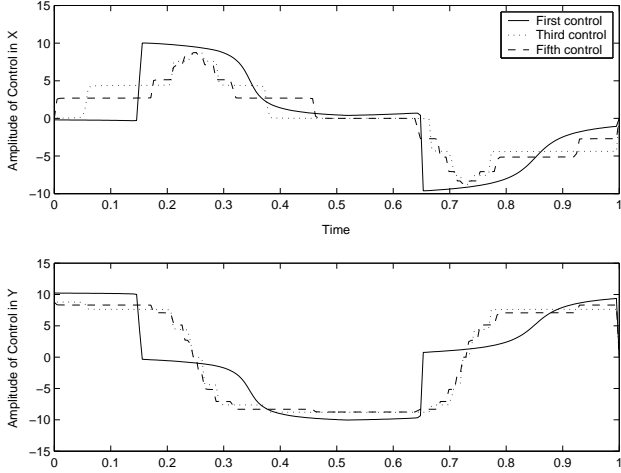


Fig. 5 Control for different circle approximations

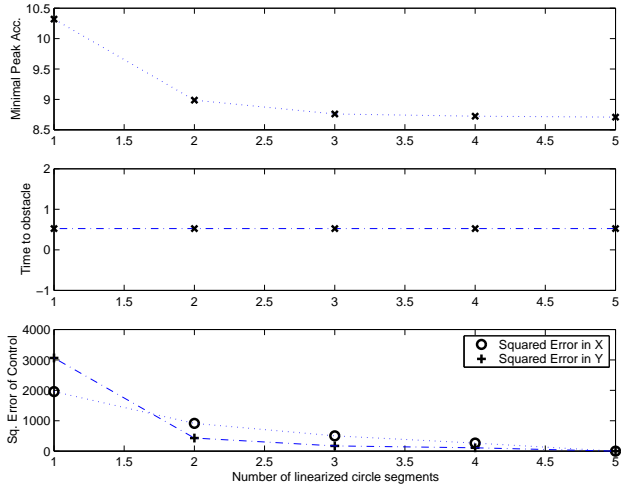


Fig. 6 M , T_{obs} and error for different n

Table 1 Result of general problem formulation

n	M_{min}	T_{obs}	$\Delta \vec{u}_x$	$\Delta \vec{u}_y$
1	10.4224	0.5235	152.69	238.4
2	9.0234	0.5235	73.67	37.3
3	8.7964	0.5235	35.78	12.57
4	8.7671	0.5235	12.37	7.29
5	8.7451	0.5235	≈ 0	≈ 0

TWO-LINK MANIPULATOR

As the problem is formulated in Cartesian space, the two-link manipulator configuration is mapped into the

CM space. For the given initial and final conditions $\vec{x}_0 = \begin{pmatrix} 2.22 \\ 0 \\ -2.693 \\ 0 \end{pmatrix}$, $\vec{x}_{T_f} = \begin{pmatrix} 7.0394 \\ 0 \\ 0.5228 \\ 0 \end{pmatrix}$ the center of mass configuration becomes

$$x_{CM0} = \begin{pmatrix} 2.5 \\ 0 \\ -0.2 \\ 0 \end{pmatrix}, x_{CM T_f} = \begin{pmatrix} 3.45 \\ 0 \\ 1.2 \\ 0 \end{pmatrix}. d_{obs} \text{ becomes } 2.82, h_{obs} \text{ becomes } 1.3. \text{ Figure 7 shows}$$

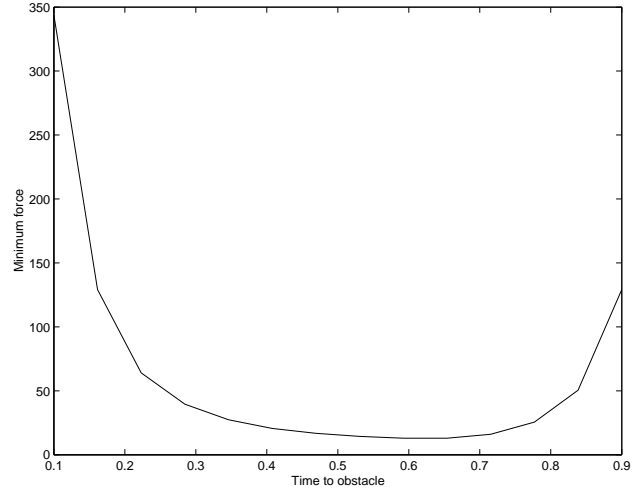


Fig. 7 M_{min} over T_{obs}

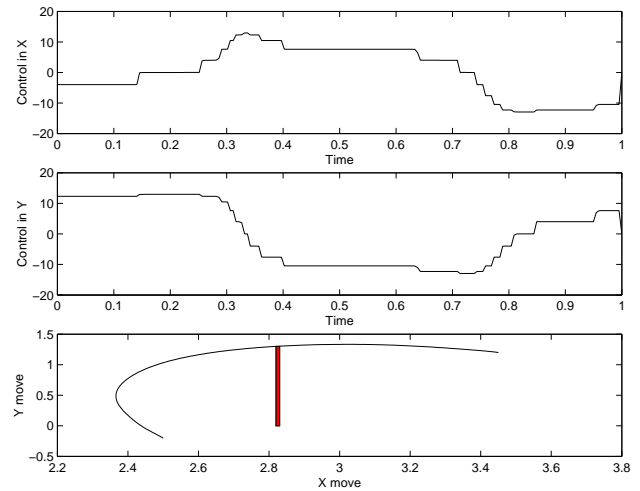


Fig. 8 Control input for robot movement

the minimal input magnitude M over the several T_{obs} . Figure 8 shows the trajectory of the control input and the $x - y$ movement in the CM space. The resulting trajectory in the Cartesian space is presented in figure 9 and the corresponding angles in figure 10.

DISCUSSION

PROBLEMS OF CM SPACE

The transformation of the robot from the Cartesian space to CM space is unique. The transformation of

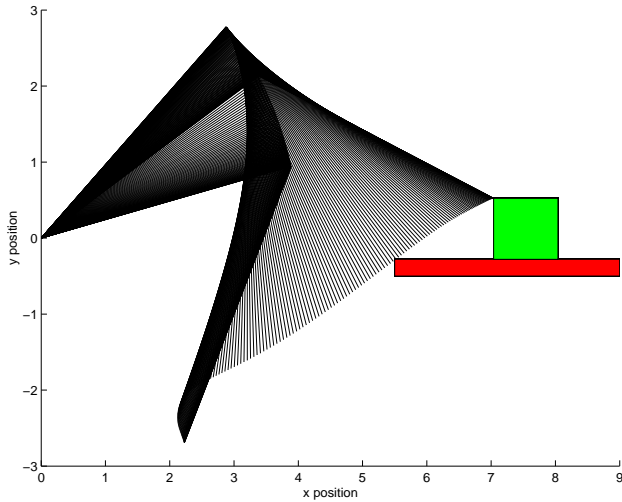


Fig. 9 Robot movement

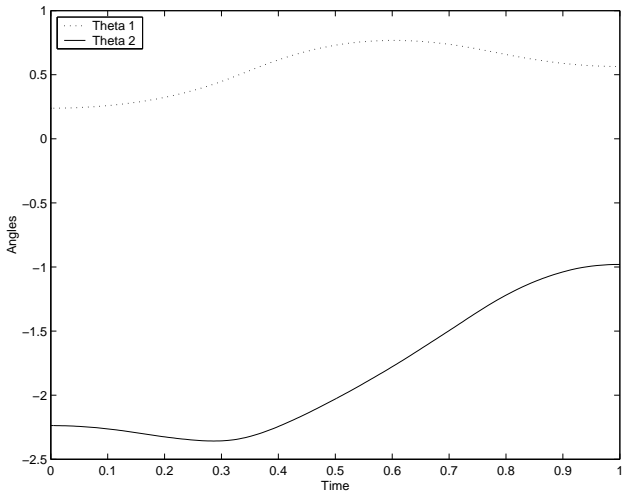


Fig. 10 Robot angles

an obstacle, however, is not unique as a robot can have different configurations of its links (e.g. two-link manipulator: elbow-up, elbow-down). The reverse transformation of the center of mass of the robot has to take non-uniqueness into account because of the different link configurations. Since the robot can change its configuration, e.g. from elbow-up to elbow-down, during one trajectory by touching one of the boundaries in the center-of-mass space it is not possible to evaluate different configurations separately. Rather candidate paths have to be found out by observation of the problem. Doggett et al. provide a further exploration of this issue.

CIRCLE APPROXIMATION

Due to the approximation of the circle by linear curves, errors occur. Here, the maximum error occurs where distance between a linear curve and the circle is maximal. If ϕ is the angle of one segment, the maximal error is the distance between the point on the circle and the point on the approximation which

corresponds to an angle of $\frac{\phi}{2}$. Figure 11 shows the relationship of the percent error and the number of segments n in one quadrant.

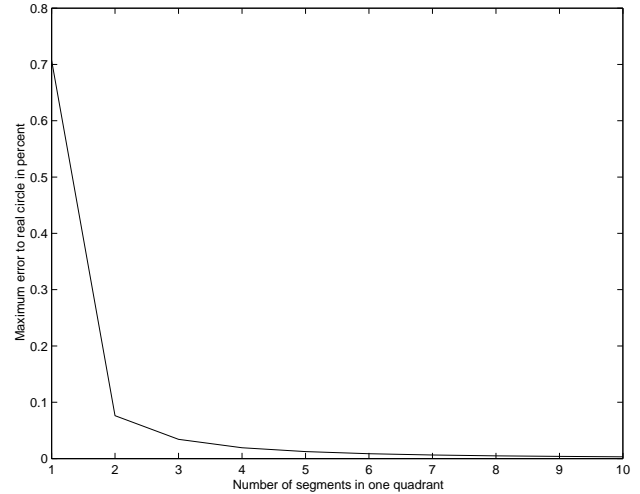


Fig. 11 Circle error

CIRCLE LINEARIZATION

As the results for the first problem show, T_{obs} hardly depends on n (table 1). The control profile for each coordinate direction changes from bang-bang control to a smooth control. This is due to the fact that there are more vertices of the approximating polygon which lie on the circle. One of the most important properties is that the squared error for both control inputs ($x-y$ directions) between two following control profiles for different n approaches 0 for higher n . This indicates convergence to a certain solution. As the circle is approximated more precise by choosing a large n , the maximum error between the approximation and the real circle decreases. The computational costs, however, increase significantly. Therefore, a good compromise between accuracy and computations has to be found.

SPLITTING OF T_{obs}

The error of the solution depends directly on the magnitude of Δt . Since the minimum M depends on T_{obs} , we get closer to the true minimum for small Δt . This means, the interval $[T, T_f]$ is examined more carefully. The accuracy, once again, is traded off versus a large number of computations. That's why a good initial guess of T_{obs} is very important. Since T_{obs} hardly changes with n , a good initial guess can be achieved by solving the problem for $n = 1$ which needs few computations. The so found T_{obs} provides a good starting point for further refinements.

APPLICATION FOR GENERAL PROBLEMS

At first glance, the problem studied seems to be a very restrictive and special example, not usable for general purposes. However, a movement from an initial point to a final point in the CM space with arbi-

trary shaped obstacles can always be transformed in a problem of the kind presented: The straight line between initial and final points will be defined as the x axis done by a displacement of the origin and a rotation of the coordinate axes. This transformation also changes the shape of the obstacle. Then, the critical points of an obstacle can be approximated by vertical lines as seen in figure 1 (multiple restrictions are possible).

CONCLUSION

An iterative approach to minimize the peak base reaction forces during $2D$ maneuvers while avoiding obstacles has been developed and tested. A linear programming technique was applied to the optimal control problem by linearization and discretization of the continuous model and the constraints. All test problems were transformed into the CM space in order to simplify calculations.

Results have shown that the method works within an acceptable accuracy. However, more precision has to be paid by higher computational effort. Since an upper bound for the error is known, computational effort and accuracy can be balanced by the user.

As a main advantage, the developed method is able to find the *global* minimum of the peak base reaction force by using an iterative technique.

REFERENCES

- ¹*Global minimization of Robotic Manipulator Base Reaction Forces*, W.R. Doggett, Bell & Howell, 2000
- ²*Multiple center of mass space images of single objects and their impact on path planning*, W.R. Doggett, W.C. Messner, Jer-Nan Juang, The international Journal of Robotics Research, Sept. 2000, vol. 19, no. 9, pp. 848-856
- ³*Computer-aided motion planning for satellite mounted robots*, V. Schulz, R. Logman, H.G. Bock, Math. Methods Appl. Sci., vol. 21, pp. 733-755, 1998
- ⁴*Optimal path planning for satellite mounted robot manipulators*, V. Schulz, R. Logman, H.G. Bock, Math. Methods Appl. Sci., vol. 82, pp. 311-329, 1993
- ⁵*Planning paths through singularities in the center of mass space*, W.R. Doggett, W.C. Messner, Jer-Nan Juang, Proc. AIAA Guidance and Control Conf., Aug. 10-12, 1998, AIAA 98-4127
- ⁶*Global minimization of the robot base reaction force during 3-D maneuvers*, W.R. Doggett, W.C. Messner, Jer-Nan Juang, IEEE Transactions on Robotics and Automation, Dec. 2000, vol. 16, no. 6, pp. 700-711
- ⁷*On computing the global time-optimal motions of robotic manipulators in the presence of obstacles*, Z. Shiller, S. Dubovsky, IEEE Transactions on Robotics and Automation, Dec. 1991, vol. 7, no. 6, pp. 785-797

⁸*Time-optimal motion planning of skid-steer mobile robots in the presence of obstacles*, M. Yamamoto, M. Iwamura, A. Mohri, Proc. IEEE Intl. Conference on Intelligent Robots and Systems, Oct. 1998

⁹*Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles*, M. Yamamoto, M. Iwamura, A. Mohri, Proc. IEEE Intl. Conference on Robotics and Automation, May 1999

¹⁰*Planning and obstacle avoidance for mobile robots*, E. Papadopoulos, I. Poulakakis, Proc. IEEE Intl. Conference on Robotics and Automation, May 2001

¹¹*Design and motion planning for a zero-reaction manipulator*, E. Papadopoulos, A. Abu-Abed, IEEE International Conference on Manipulators and Automation, San Diego, CA, vol 2, pp. 1554-1559