

**AN OPTIMAL CONTROL STRATEGY  
FOR COLLISION AVOIDANCE  
OF MOBILE ROBOTS IN  
NON-STATIONARY ENVIRONMENTS**

*NAGW-1333*

by

K.J. Kyriakopoulos and G.N. Saridis

Rensselaer Polytechnic Institute  
Electrical, Computer, and Systems Engineering  
Troy, New York 12180-3590

February 1991

**CIRSSE REPORT #83**

# An Optimal Control Strategy for Collision Avoidance of Mobile Robots in Non-stationary Environments

K.J.Kyriakopoulos and G.N.Saridis

NASA Center for Intelligent Robotic Systems for Space Exploration  
Robotics and Automation Laboratories

Rensselaer Polytechnic Institute  
Troy, NY 12180-3590

## abstract

An optimal control formulation of the problem of collision avoidance of mobile robots in environments containing moving obstacles is presented. Collision avoidance is guaranteed if the minimum distance between the robot and the objects is nonzero. A nominal trajectory is assumed to be known from off-line planning. The main idea is to change the velocity along the nominal trajectory so that collisions are avoided. Furthermore, time consistency with the nominal plan is desirable. A numerical solution of the optimization problem is obtained. Simulation results verify the value of the proposed strategy.

## 1 Introduction

The problem of "moving a mobile robot or a manipulator from an initial position of the workspace to a final one, avoiding the stationary obstacles of the environment and optimizing over certain criteria while satisfying the kinematic and dynamic constraints of the robot", is well known to be formidable due to its NP-complete computational complexity [Can86].

Undoubtedly, the more complex version of the problem where moving obstacles are present (figure 1) in the environment is even more difficult and computationally expensive. The problem of treating moving objects has been stated as early as 1984 [FH84] and usually ad hoc solutions were given [Le89] [Tou86]. Reif and Sharir [RS85] gave an algorithmic solution to the problem but they were restricted to some categories of shapes of objects. and their approach is not suitable for an on-line implementation. On the other hand, Kant and Zucker [KZ84], [KZ86], [KZ88], used the decomposition of the motion planning problem to the *find-path*, and *move-along-path* problems, they propose that the avoidance of moving obstacles can be done by adjusting the motion along the geometric path. The same approach was adopted in [WJ88], and recently in [GE90]. The basic idea of this approach is utilized in this work. Our scheme is more general and complete in the sense that the dynamic model of the robot is used, the objects are modeled as convex polyhedra and, in addition to collision avoidance, time consistency with the nominal plan is sought. Lately search based

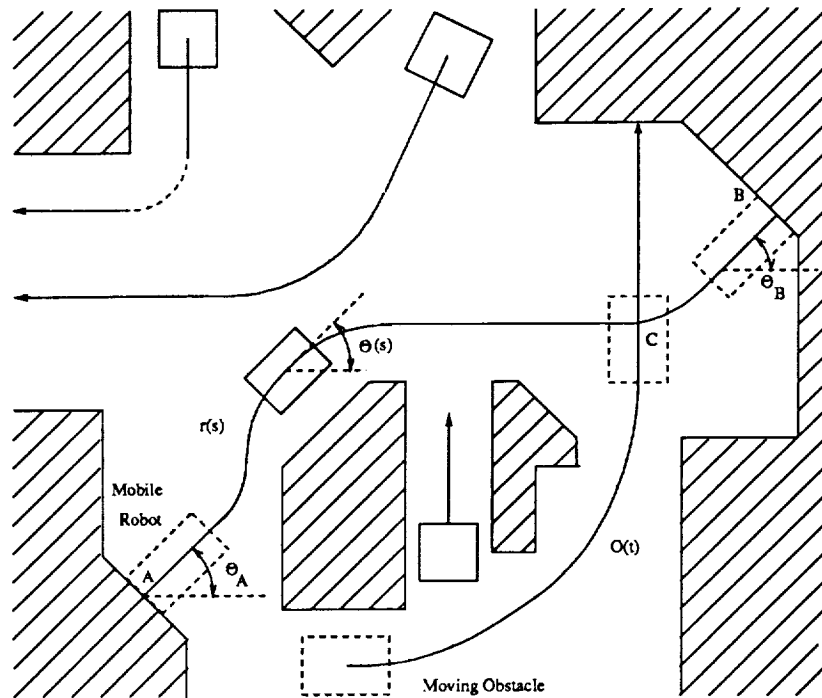


Figure 1: Environment with a mobile robot and multiple moving obstacles

approaches for solving the above problem have also been presented in [FS90],[SLG90].

To facilitate a fast solution a hierarchical decomposition has been proposed [KZ88], and adopted and extended in our work. The problem is divided in two:

- Off-line Path and Motion Planning, and
- On-line Motion Replanning.

In the **off-line stage**, two problems are solved: First, path planning, the “find path” problem, ie the search for a connected curve  $r(s) = [p_x(s) p_y(s)]^T$  and an associated orientation function  $\theta(s)$ , where  $s$  is the trajectory parameterization variable, in the robot’s workspace, connecting the initial and target points without colliding with the stationary objects while satisfying certain criteria. Second, motion planning, the problem of how to move along this path without violating the kinematic and dynamic constraints of the robot, and some performance criterion (e.g time) is minimized.

The subject of this paper is the development of an algorithm for the **on-line stage**. This algorithm has to act in a supervisory mode during motion execution in order to avoid collisions with moving obstacles with which collision is predicted based on sensory input. The new plan must satisfy the dynamic constraints and stay as close as possible to the nominal plan.

In our previous developments [KS90b] [KS91] the Minimum Interference Strategy (MIS) was proposed to give fast, feasible and close to optimal solutions. In this work, the Optimal Control Strategy (OCS) is proposed to give optimal solutions of the problem at hands.

In section 2 the preliminary definitions and the problem are stated. The theoretical analysis and the presentation of OCS is done in section 3. Finally, in section 4 simulation results and suggestions for future research are presented.

## 2 Definitions - Assumptions - Modeling

Collision avoidance is guaranteed if the distance  $d(s, t)$  [GJ85] between the robot and the object is greater than a safety positive constant  $d^o$ , i.e

$$d(s, t) = \min_{i,j} \{ \|z_i - z_j\| : z_i \in C_r(s), z_j \in C_o(t) \} \geq d^o \quad \forall s, t \quad (1)$$

where

$$C_r(s) = \{x/x \in \mathbb{R}^3 \ni A_r \cdot R_r^{-1}(s) \cdot x \leq b_r - A_r \cdot R_r^{-1}(s) \cdot T_r(s), \quad A_r \in \mathbb{R}^{m \times 3}, b_r \in \mathbb{R}^m\}, \quad (2)$$

$$C_o(t) = \{y/y \in \mathbb{R}^3 \ni A_o \cdot R_o^{-1}(t) \cdot y \leq b_o - A_o \cdot R_o^{-1}(t) \cdot T_o(t), \quad A_o \in \mathbb{R}^{l \times 3}, b_o \in \mathbb{R}^l\}, \quad (3)$$

are **convex polyhedra** representing the convex hulls of the mobile robot and the moving obstacle respectively.  $(A_r, b_r)$  and  $(A_o, b_o)$  are the parameters that define the convex polyhedron description of the robot and the object respectively, with respect to their fixed coordinate frame.  $R_r, R_o, T_r, T_o$  represent the rotation and translation of the frames of the robot and the object with respect to the world frame. A computationally efficient approach to estimate  $d(s, t)$  and predict the collision time  $t_c$  under uncertain input from sensing devices is presented elsewhere [KS90a]. The collision time is defined by

$$t_c = \inf_{t \in [t_0, t_0 + T_h]} \{t/d(t) = 0\} \quad (4)$$

where  $t_0$  is the present time, and  $T_h$  is the time horizon of interest.

The proposed approach works if the following assumption is satisfied.

**Temporary Obstruction Assumption:** The mobile robot moving along path  $r(s)$  can only be obstructed during a bounded amount of time i.e the moving object is assumed not to permanently stay on, or move parallel to  $r(s)$ .

The problem of planning the motion of the mobile robot such that it avoids the moving obstacles while minimizing deviations from the nominal final time can be posed as an optimization problem. The mathematical statement of the problem and the derivations of the dynamic model and the constraints has been presented in [KS90b] [KS91].

System

$$x'(s) = A(x(s)) + B(x(s)) \cdot u(s) \quad (5)$$

where  $(\cdot)' = \frac{d(\cdot)}{ds}$ ,  $x(s) = [t(s) \ v(s)]^T$ , with  $v(s) = \frac{ds}{dt}(s)$ ,

$$A(x) = \begin{bmatrix} \frac{1}{v(s)} \\ \frac{-I \cdot f(s) \cdot f'(s)}{m + I \cdot f^2(s)} \cdot v(s) \end{bmatrix} \quad (6)$$

and

$$B(x) = \begin{bmatrix} 0 \\ \frac{1}{m+I \cdot f^2(s)} \cdot \frac{1}{v(s)} \end{bmatrix} \quad (7)$$

### Initial-Final Conditions

$$x(s_0) = [0 \ v_0]^T \quad x(s_f) = [\text{free} \ v_f]^T \quad (8)$$

### input constraints

$$-U_2 \leq u(s) \leq U_1 \quad (9)$$

### State Constraints

$$0 < \epsilon \leq v(s) \leq v_{stab}(s) \quad (10)$$

### Collision Avoidance Constraints

$$d_0 - d(s, t) \leq 0 \quad (11)$$

### Performance Criterion

$$J = \frac{1}{2}(t(s_f) - T)^2 \quad (12)$$

where  $s$  is the trajectory parameterization variable,  $t(s), v(s)$  are the time and the velocity when mobile robot is at point  $s$  of the trajectory.  $I$  is the moment of inertia around the center of translation,  $f(s)$  is the curvature of the cartesian trajectory  $r(s)$ ,  $m$  is the mass of the mobile robot and  $u$  is the input (force) which is bounded. The velocity  $v(s)$  is bounded at every instant to guarantee the nonslipping conditions [K.K91].  $T$  is the final time of the nominal plan (i.e  $t_n(s_f) = T$ ). Notice that the criterion is simply convex, but not strictly. This means that the optimal solution is not necessarily unique.

## 3 Optimal Control Strategy (OCS)

The problem is posed as an optimal control problem with constraints on the states (hard) and the control (soft). The analytic solution of such problems is a formidable problem. Thus a numerical solution is sought. However in order to investigate the convergence performance of such an algorithm, the properties of the feasible control space should be investigated.

### 3.1 Feasible Controls Space

The feasible controls space is defined as

$$\mathcal{U} \triangleq \left\{ u \in L^2(0, s_f, \mathfrak{R}) / \exists (8 - 11) \text{ are satisfied} \right\} \quad (13)$$

Its convexity properties should be investigated for numerical convergence purposes.

Obviously (9) is convex since for all inputs  $u_1, u_2 \in L^2(0, s_f, \mathfrak{R})$  that are bounded (i.e  $u_1(\sigma), u_2(\sigma) \in [-U_2, U_1], \sigma \in [0, s_f]$ ) it is obvious that its convex combination  $u = \lambda \cdot u_1 + (1 - \lambda) \cdot u_2$  is bounded (i.e  $u(\sigma) \in [-U_2, U_1], \sigma \in [0, s_f]$ ).

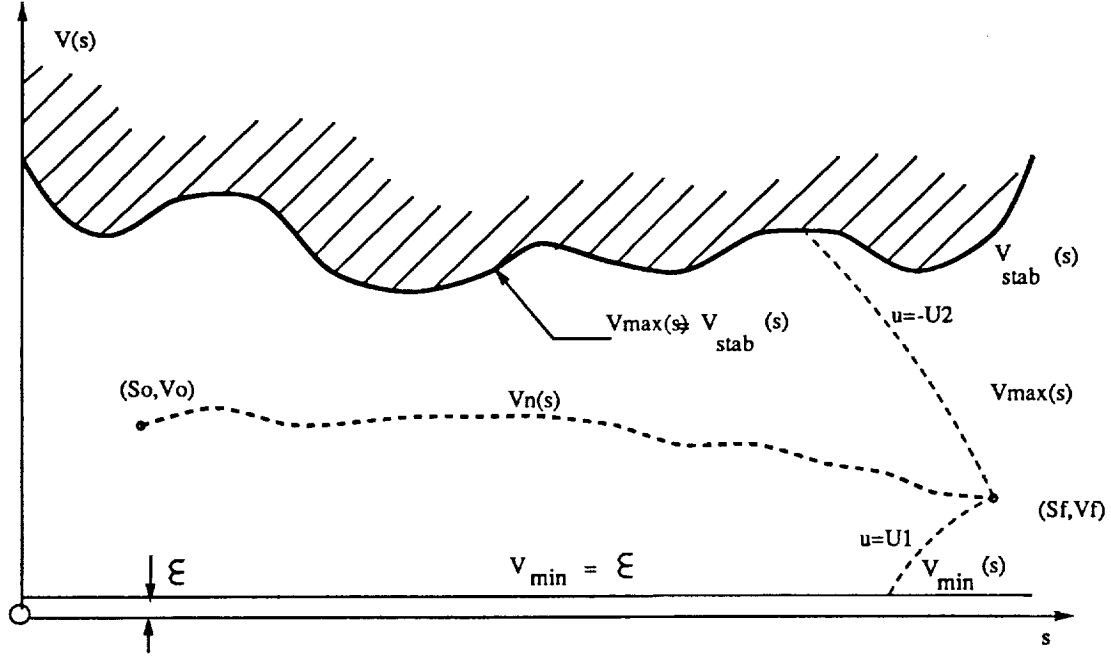


Figure 2: Feasible Velocities Space

The final equality constraint (8) can be imposed using inequalities. Consider as upper and lower velocity bounds the two velocity arcs starting from  $(s_f, v_f)$  and going backwards with input  $-U_2, U_1$  until they meet  $v_{stab}(s), v(s) = \epsilon$  respectively. Thus  $v_{max}(s)$  and  $v_{min}(s)$  are constructed.

This is demonstrated in figure 2. Thus, a new set of inequalities

$$0 < v_{min}(s) \leq v(s) \leq v_{max}(s) \quad (14)$$

is used to satisfy (8,10). Due to the assumption about motion on a horizontal plane, the total energy  $\mathcal{E}$  of the mobile robot at every instant, is equal to its Kinematic energy. Thus,

$$\mathcal{E}(s, v) = \frac{1}{2} (m + I \cdot f^2(s)) \cdot v^2(s) \quad (15)$$

where  $f(s)$  is the curvature of the path  $r(s)$ . Integrating the second of the state equations the kinematic energy at a point  $s$  is given by

$$\frac{1}{2} (m + I \cdot f^2(s)) \cdot v^2(s) = \frac{1}{2} (m + I \cdot f^2(s_0)) \cdot v_0^2(s_0) + \int_{s_0}^s u(\sigma) \cdot d\sigma \quad (16)$$

where  $u(\sigma) \in L^2(0, s_f, \mathfrak{R})$ . Thus,

$$v^2(s) = \frac{(m + I \cdot f^2(s_0))}{(m + I \cdot f^2(s))} \cdot v_0^2(s_0) + \frac{2 \int_{s_0}^s u(\sigma) \cdot d\sigma}{(m + I \cdot f^2(s))} \quad (17)$$

If velocities  $v_1(s), v_2(s)$

$$v_1^2(s) = \frac{(m + I \cdot f^2(s_0))}{(m + I \cdot f^2(s))} \cdot v_0^2(s_0) + \frac{2 \int_{s_0}^s u_1(\sigma) \cdot d\sigma}{(m + I \cdot f^2(s))} \quad (18)$$

$$v_2^2(s) = \frac{(m + I \cdot f^2(s_0))}{(m + I \cdot f^2(s))} \cdot v_0^2(s_0) + \frac{2 \int_{s_0}^s u_2(\sigma) \cdot d\sigma}{(m + I \cdot f^2(s))} \quad (19)$$

satisfy (14), then velocity  $v(s)$

$$v^2(s) = \frac{(m + I \cdot f^2(s_0))}{(m + I \cdot f^2(s))} \cdot v_0^2(s_0) + \frac{2 \int_{s_0}^s u(\sigma) \cdot d\sigma}{(m + I \cdot f^2(s))} \quad (20)$$

where  $u(\sigma) = \lambda \cdot u_1(\sigma) + (1 - \lambda) \cdot u_2(\sigma)$   $\sigma \in [0, s]$ , satisfies

$$v^2(s) = \lambda \cdot v_1^2(s) + (1 - \lambda) \cdot v_2^2(s) \quad (21)$$

and therefore it satisfies (14). Therefore (14) is convex.

Consider the sets

$$D \triangleq \{(s, t) / d(s, t) - d^0 \leq 0\} \quad (22)$$

$$D_s \triangleq \{s / \exists t \ni (s, t) \in D\} \quad (23)$$

For a specific  $s$ , the **Collision Times Set** (time instants where collision is possible if the robot is at position  $s$ ) is defined as

$$C_t(s) \triangleq \{t / \ni (s, t) \in D\} \quad (24)$$

Since  $d(s, t)$  is continuous w.r.t both arguments, it can be easily proved that if a pair  $(s_c, t_c)$  exists such that  $d(s_c, t_c) = 0$  then  $C_t(s)$  is found to be non-empty and compact  $\forall s \in D_s$ . Thus the following definitions are meaningful

$$t_l(s) \triangleq \min C_t(s) \quad (25)$$

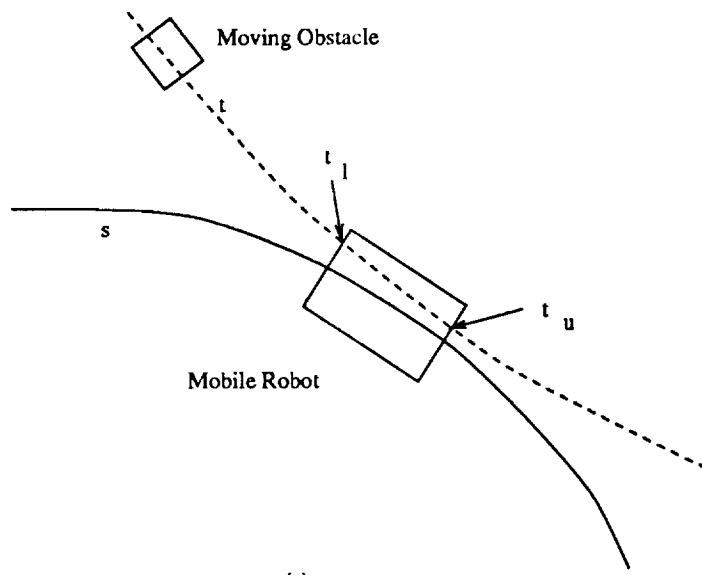
$$t_u(s) \triangleq \max C_t(s) \quad (26)$$

Notice that

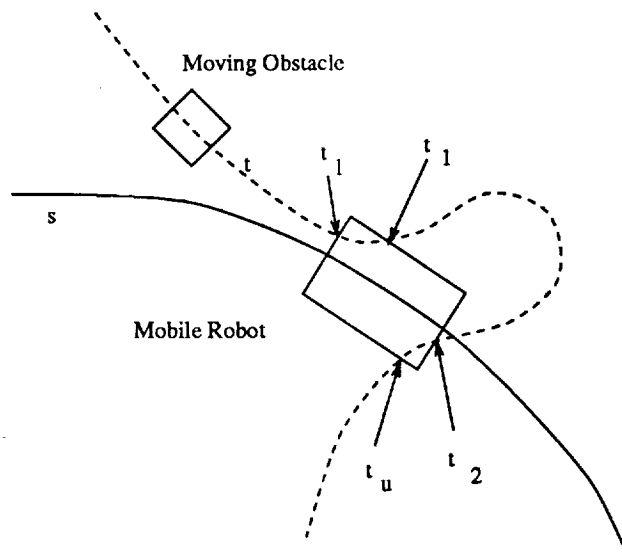
$$\left. \begin{array}{l} t(s) \geq t_u(s) \forall s \in D_s \\ \text{or} \\ t(s) \leq t_l(s) \forall s \in D_s \end{array} \right\} \implies d_0 - d(s, t) \leq 0 \quad (27)$$

showing the sufficiency of those collision avoidance conditions. However, the necessity is not true in general. This is demonstrated in figure 3(b). There it is obvious that  $C_t(s_0) = \{t / t \leq t_l \vee t_1 \leq t \leq t_2 \vee t \geq t_u\}$ .

For safety purposes we are not interested in the interval  $[t_1, t_2]$ . Furthermore, in practice, the conic section type of obstacle trajectories that result from the used estimation algorithm [KS90a] will not give such intervals. Thus, it becomes obvious that the feasible space is



(a)



(b)

Figure 3: Examples of Motion of Obstacles



decomposed in terms of the collision avoidance constraint in two: the first is the **space of decelerating plans** where

$$t(s) \geq t_u(s) \forall s \in D_s \quad (28)$$

and the second is **space of accelerating plans** where

$$t(s) \leq t_l(s) \forall s \in D_s \quad (29)$$

Therefore we will seek solutions of the optimal control problem in the following feasible control spaces:

Deceleration Space

$$\mathcal{U}_{dec} \triangleq \left\{ u \in L^2(0, s_f, \mathfrak{R}) / \ni (9, 14, 28) \text{ are satisfied} \right\} \quad (30)$$

Acceleration Space

$$\mathcal{U}_{acc} \triangleq \left\{ u \in L^2(0, s_f, \mathfrak{R}) / \ni (9, 14, 29) \text{ are satisfied} \right\} \quad (31)$$

Consider the following lemma:

**Lemma .1** *If*

- a)  $f(\sigma), g(\sigma) \in \mathcal{C}[0, 1]$ ,
  - b)  $f(\sigma), g(\sigma) > 0 \quad \forall \sigma \in [0, 1]$
  - c)  $f, g$  not effectively proportional and
  - d)  $s \in [0, 1]$  fixed,
- then  $M(\lambda) : [0, 1] \rightarrow \mathfrak{R}$  defined by:

$$M(\lambda) \triangleq \int_0^s \frac{1}{\sqrt{\lambda \cdot f + (1 - \lambda) \cdot g}} d\sigma \quad (32)$$

is convex w.r.t. its argument  $\lambda$ .

Proof: The proof is given in appendix. ■

An input  $u_1(s) \in \mathcal{U}_{acc}$  results to  $v_1(s)$  such that

$$t_1(s) = \int_0^{s_f} \frac{1}{v_1(\sigma)} \cdot d\sigma \leq t_l(s) \quad (33)$$

and  $u_2(s) \in \mathcal{U}_{acc}$  results to  $v_2(s)$  such that

$$t_2(s) = \int_0^{s_f} \frac{1}{v_2(\sigma)} \cdot d\sigma \leq t_l(s) \quad (34)$$

Their convex combination  $u(\sigma) = \lambda \cdot u_1(\sigma) + (1 - \lambda) \cdot u_2(\sigma)$  results to  $v(s)$  satisfying (21) i.e

$$v(s) = \sqrt{\lambda \cdot v_1^2(s) + (1 - \lambda) \cdot v_2^2(s)} \quad (35)$$

Because of Lemma .1 it is deduced that

$$t(s) = \int_0^{s_f} \frac{1}{v(\sigma)} \cdot d\sigma \leq \min\{t_1(s), t_2(s)\} \stackrel{(33,34)}{\implies} t(s) \leq t_l(s) \quad (36)$$

Therefore  $\mathcal{U}_{acc}$  is convex.

Efforts to prove the same for  $\mathcal{U}_{dec}$  have not led to any result.

### 3.2 Numerical Implementation

A closed form solution of the optimal control problem at hands is a formidable task. Therefore a numerical solution was sought. The nonlinear hard constraints (11,14) that can be written in the vector form

$$G(x(s), s) \geq 0 \quad G(.,.) \in \mathfrak{R}^3 \quad (37)$$

complicate the numerical solution. The approach of [Mar73] was used to approximate them with equivalent soft constraints  $g(.,.,.) \in \mathfrak{R}^3$  that satisfy

$$g(x(s), u(s), s) = C(x, s) + \mathcal{D}(x, s) \cdot u(s) \geq G(x(s), s) \geq 0 \quad \forall s > s_0 \quad (38)$$

In appendix the approach of [Mar73] is presented.

If we incorporate the control input constraints into  $g(.,.,.)$  then  $g(.,.,.) \in \mathfrak{R}^5$ .

To enhance numerical convergence the penalty functions approach was adopted. Thus, criterion (12) was changed to

$$J_1 = \frac{1}{2}(t(s_f) - T)^2 + \int_{s_0}^{s_f} \left( \sum_{j=1}^5 r_j \cdot \frac{1}{C_j(x(s), s) + \mathcal{D}_j(x(s), s) \cdot u(s)} \right) \cdot ds \quad (39)$$

where  $r_j \ j = 1, \dots, 5$  are very small positive constants.

Several attempts to solve the problem with shooting and matching techniques proved to be futile. The only approach that proved to be successful was the **Nonlinear Programming** approach.

The numerical solution of the above problem requires the discretization of the domain  $[0, s_f]$  in  $N$  steps of uniform length  $\Delta s = \frac{s_f}{N}$ , creating a grid of  $N + 1$  points  $\{0, s_1, \dots, s_N = s_f\}$ . The corresponding  $N$  control inputs  $\{u_0, u_1, \dots, u_{N-1}\}$ ,  $N$  time instants  $\{t_1, t_2, \dots, t_N\}$  and  $N$  velocities  $\{v_1, v_2, \dots, v_N\}$ , where  $u_i = u(s = i \cdot \Delta s)$ ,  $t_i = t(s = i \cdot \Delta s)$ ,  $v_i = v(s = i \cdot \Delta s)$ , are the unknowns of the problem.

The nonlinear programming problem is stated, in the form:

$$\begin{aligned} \min_{x \in \mathfrak{R}^{3 \cdot N}} \quad & F(x) \\ \text{s.t} \quad & l_x \leq x \leq r_x \\ & A \cdot x = 0 \\ & c_t(x) = 0 \\ & c_d(x) \leq 0 \end{aligned} \quad (40)$$

where

$x = [u_0, u_1, \dots, u_{N-1}, t_1, t_2, \dots, t_N, v_1^2, v_2^2, \dots, v_N^2]^T$  is the vector of unknowns (Notice that the velocity components of this vector are the squares of the velocities. This is going to be justified later), and

$F(x) = \frac{1}{2}(t_N - T)^2 + \sum_{i=0}^{N-1} \left( \sum_{j=1}^5 r_j \cdot \frac{1}{C_j(x(s_i), s_i) + \mathcal{D}_j(x(s_i), s_i) \cdot u(s_i)} \right) \cdot (s_{i+1} - s_i)$ , where  $T$  is the total time in the nominal plan.

The bounds of the vector of unknowns  $x$  are :

For  $u_i$ :

$$-U_2 \leq u_i \leq U_1 \quad i = 0, 1, \dots, N-1 \quad (41)$$

which actually is constraint (9).

For  $v_i^2$ :

$$v_{min}^2(s_i) \leq v_i^2 \leq v_{max}^2(s_i) \quad i = 1, \dots, N \quad (42)$$

representing (14).

The linear equality constraints ( $A \cdot x = 0$ ) are

$$(m + I \cdot f^2(s_i)) \cdot v_i^2 - (m + I \cdot f^2(s_{i-1})) \cdot v_{i-1}^2 - 2 \cdot (s_i - s_{i-1}) \cdot u_i = 0 \quad i = 1, \dots, N \quad (43)$$

representing the dynamics of the system as given by the second state equation in (5). Notice that because we use  $v_i^2$  in the vector of unknowns  $x$ , this is a linear constraint otherwise, it would be a nonlinear one.

The nonlinear equality constraints ( $c_t(x) = 0$ ) are

$$t_{i+1} - t_i - \frac{s_{i+1} - s_i}{\frac{\sqrt{v_{i+1}^2} + \sqrt{v_i^2}}{2}} = 0 \quad i = 0, \dots, N-1 \quad (44)$$

representing the first of the state equations (5).

Finally, the nonlinear inequality constraints ( $c_d(x) \leq 0$ ) are

$$d_0 - d(s_i, t_i) \leq 0 \quad i = 1, \dots, N \quad (45)$$

which actually represent the collision avoidance constraint (11).

A **sequential quadratic programming method** (SQP), (see for example [Pow83]), in which the search direction is a solution of a quadratic programming problem was adopted. The NAG Fortran Library Routine E04UCF actually implements the above algorithm and it was used to obtain the numerical results of the case study chapter.

Notice that SQP is a gradient search method which is a local method. If the initial guess  $x^0$  of the algorithm corresponds to an input vector  $u^0$  that is in  $\mathcal{U}_{acc}$  then the algorithm is going to converge to  $x^*$  corresponding to  $u^* \in \mathcal{U}_{acc}$ . Similarly, if the initial guess  $x^0$  of the algorithm corresponds to an input vector  $u^0$  that is in  $\mathcal{U}_{dec}$  then the algorithm is going to converge to  $x^*$  corresponding to  $u^* \in \mathcal{U}_{dec}$ . An efficient method to obtain initial, and very close to optimal guesses either in  $\mathcal{U}_{acc}$  or in  $\mathcal{U}_{dec}$  is provided by the Minimum Interference Strategy (MIS) [KS91]. The Accelerating MIS provides a first guess  $x^0 \in \mathcal{U}_{acc}$ , while the Decelerating MIS provides a first guess  $x^0 \in \mathcal{U}_{dec}$ .

SQP requires the derivatives of both the cost criterion and the constraints, w.r.t  $x$ . An analytical form of those derivatives can be easily obtained except for constraint (45) that involves the distance function  $d(s, t)$ . This is well known to be continuous but not continuously differentiable [GJ85]. A way to bypass this difficulty is to predict the "singularity" points where the derivative is discontinuous and interpolate locally with a sigmoid function. This approach was used in the case study and did not create any numerical problems.

The algorithm is for the Optimal Control Strategy is:

### OCS Algorithm

- STEP1:** Find an initial input guess  $u^0 = \{u_0^0, u_1^0, \dots, u_{N-1}^0\} \in \mathcal{U}_{acc}$  and the corresponding time and velocity trajectories  $t^0 = \{t_1^0, t_2^0, \dots, t_N^0\}$ ,  $v^0 = \{v_1^0, v_2^0, \dots, v_N^0\}$ . Formulate the initial guess vector  $x^0$ .
- STEP2:** Feed  $x^0$  into SQP and let it converge to  $x_{acc}^*$ , providing  $u_{acc}^*, t_{acc}^*$  and  $v_{acc}^*$ .
- STEP3:** If  $J_{acc} = (t_{acc}^*(s_N) - T)^2$  is very small, then an optimal solution has been found. **STOP**;  
Otherwise continue.
- STEP4:** Find initial input guess  $u^0 = \{u_0^0, u_1^0, \dots, u_{N-1}^0\} \in \mathcal{U}_{dec}$  and the corresponding time and velocity trajectories  $t^0 = \{t_1^0, t_2^0, \dots, t_N^0\}$ ,  $v^0 = \{v_1^0, v_2^0, \dots, v_N^0\}$ . Formulate the initial guess vector  $x^0$ .
- STEP5:** Feed  $x^0$  into SQP and let it converge to  $x_{dec}^*$ , providing  $u_{dec}^*, t_{dec}^*$  and  $v_{dec}^*$ .
- STEP6:** If  $J_{dec} = (t_{dec}^*(s_N) - T)^2$  is very small, then an optimal solution has been found. **STOP**;  
Otherwise compare  $J_{acc}$  and  $J_{dec}$  and chose optimal solution.

## 4 Simulation Results

In this section a case study is presented. A mobile robot and a moving obstacle with geometric shapes, moving in the same environment 4. The shape of both the mobile robot and the moving obstacle is rectangle with dimensions  $0.3m \times 0.52m$  and  $0.28m \times 0.28m$ , respectively. The scenario is that when the mobile robot is about to start moving, it detects the obstacle, estimates its kinematic parameters

$$\begin{aligned} x_0 &= 4.0 \text{ m} & y_0 &= 7.0 \text{ m} \\ v_x &= 0.04 \text{ m/sec} & v_y &= -0.075 \text{ m/sec} \\ a_x &= 0.094 \text{ m/sec}^2 & v_y &= -0.041 \text{ m/sec}^2 \end{aligned}$$

and predicts a collision under the current plan at  $t_c = 6.67 \text{ sec}$ .

The mobile robot has parameters:

*Mass* ( $M$ ) : 60 Kg.

*Inertia* ( $I$ ) : 32 Kg.m<sup>2</sup>

*Maximum Accelerating Force* ( $U_1$ ) : 140 N

*Minimum Decelerating Force* ( $U_2$ ) : -60 N

*Maximum Velocity* ( $V_{max}$ ) : 8 m/sec<sup>2</sup>

*Wheel-floor friction coefficient* ( $\eta$ ) :  $\sim 0.3$

has the task to go from configuration  $A$  to configuration  $B$  within  $T = 12.1753 \text{ sec}$ . An off-line path planning stage is done and a path  $r(s)$   $0 \leq s \leq s_f$  with total length  $s_f \approx 14.60 \text{ m}$  is computed. The parameters of  $r(s)$  are indicated on figure 4.

Initial and final velocities are zero ( $v_A = v_B = 0$ ).

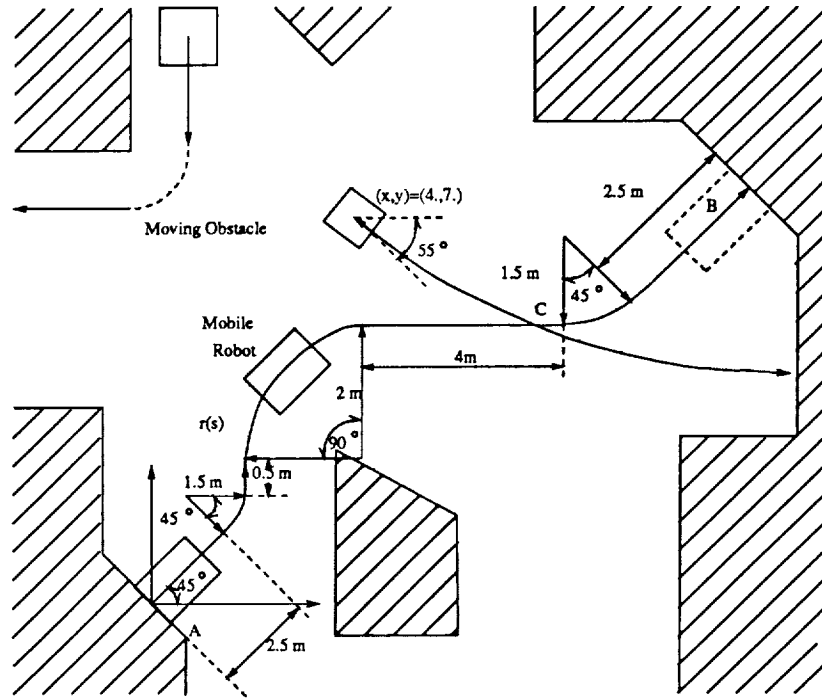


Figure 4: Environment with a mobile robot and a moving obstacle

The NAG Fortran Library Routine E04UCF was used to solve the nonlinear programming problem. The whole segment was divided in 111 segments. The first guesses required for the accelerating OCS ( $u \in \mathcal{U}_{acc}$ ) and the decelerating OCS ( $u \in \mathcal{U}_{dec}$ ), were given by AMIS and DMIS respectively [KS91].

The resulting velocity profiles from the Accelerating OCS and the Decelerating OCS are plotted with solid line on figures 5 and 6 respectively.

The time functions for Accelerating OCS ( $t_{acc}(s)$ ), Decelerating OCS ( $t_{dec}(s)$ ) and the nominal plan ( $t_{nom}(s)$ ) are plotted on figure 7. The motion time of the nominal plan is  $T = 12.1753 \text{ sec}$ , while  $T_{acc} = 12.1753 \text{ sec}$  and  $T_{dec} = 12.2782 \text{ sec}$ .

The optimal control formulation although mathematically sound suffers from computational complexity that prevents its on-line use. However the initial guess provided by MIS is very close to optimal and therefore the convergence is speeded up. Issues related to a parallel asynchronous implementation of the above algorithm for multiprocessor environments are of our immediate interest. Furthermore, the issue of sensitivity of solution to the estimates of the kinematic parameters of the moving object has to be investigated. Preliminary results show that the present algorithm can be updated very efficiently. This is a very attractive feature because the initial penalty of computation is not frequently repeated.

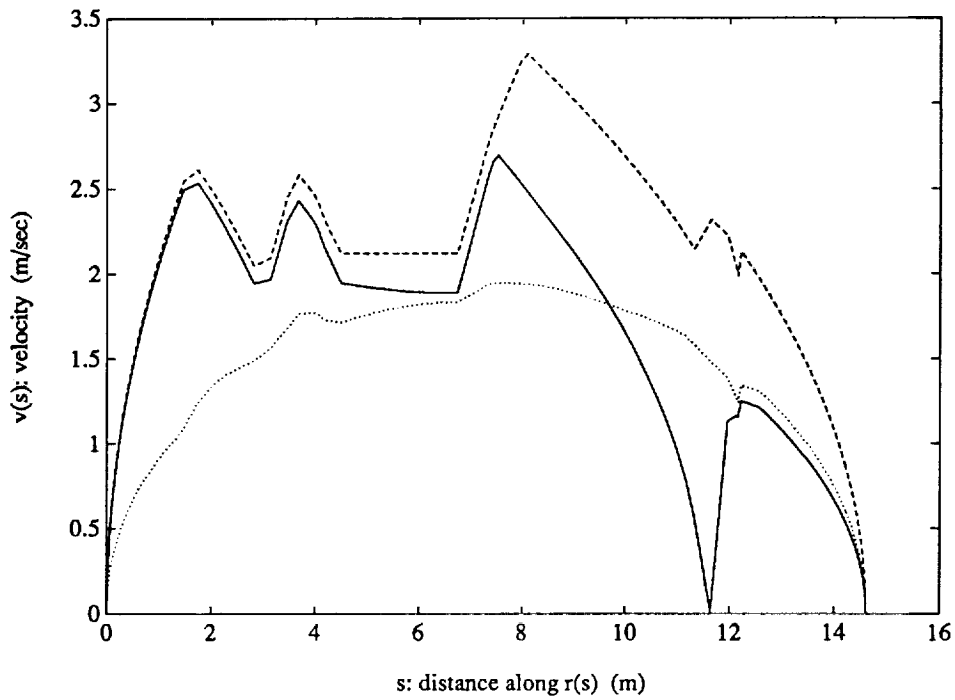


Figure 5: Velocity profile of Accelerating OCS (—), nominal velocity (....) and  $v_{max}$  (- - -)

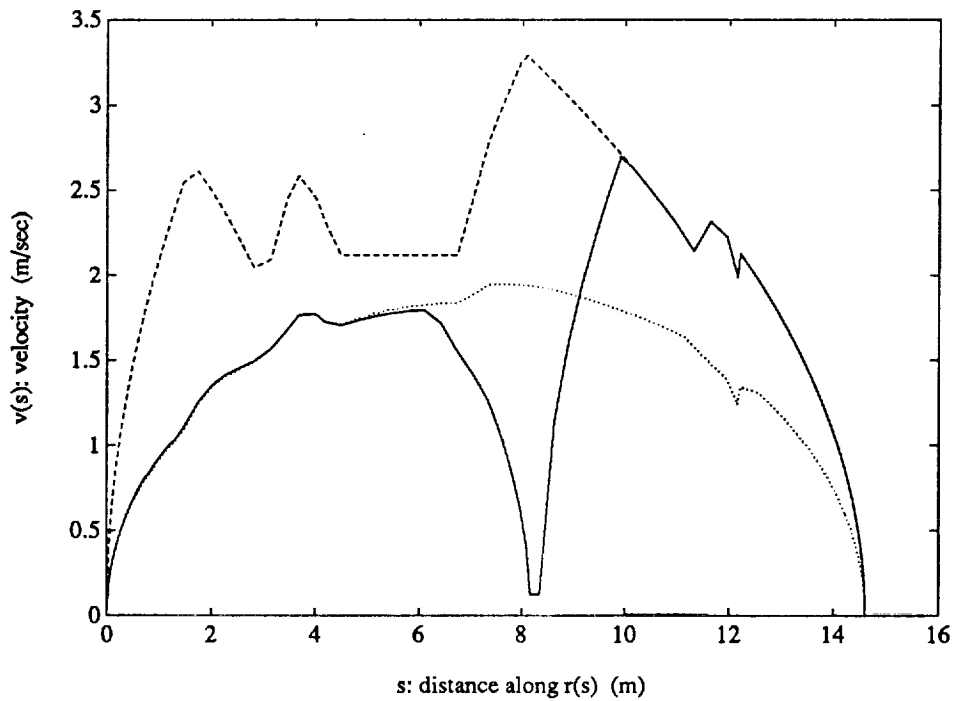


Figure 6: Velocity profile of Decelerating OCS (—), nominal velocity (....) and  $v_{max}$  (- - -)

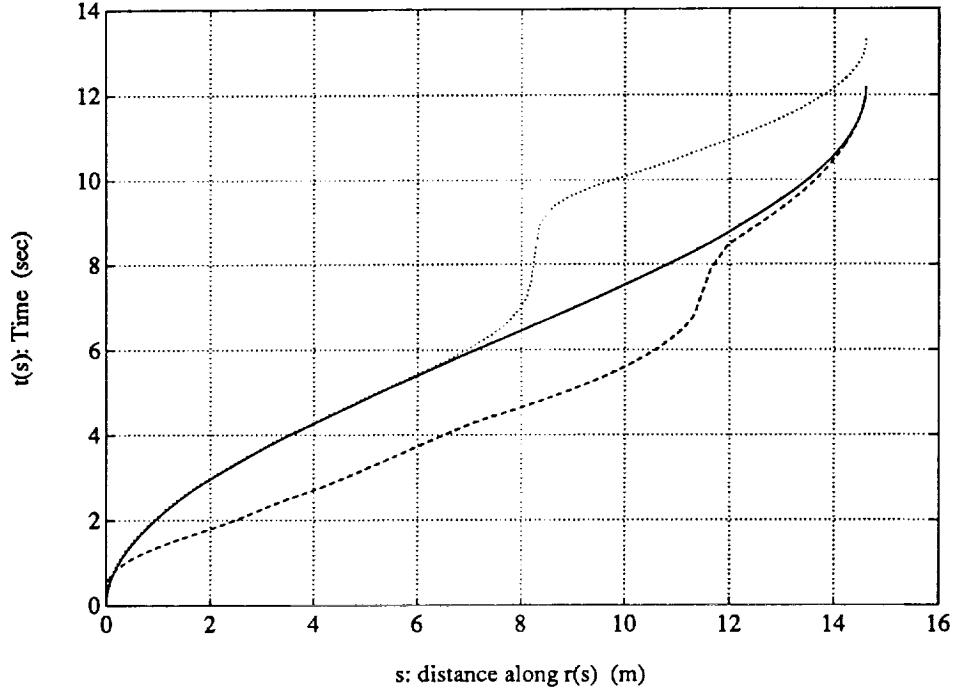


Figure 7: Time for Accel. OCS (- - -), Decel. OCS (.....) and the nominal plan (—)

## APPENDIX

### Proof of Lemma

Since  $f(\sigma), g(\sigma) \in C[0, 1]$

$$\frac{dM(\lambda)}{d\lambda} = \frac{1}{2} \int_0^s [\lambda \cdot f + (1 - \lambda) \cdot g]^{-\frac{3}{2}} \cdot (g - f) \cdot d\sigma \quad (46)$$

and

$$\frac{d^2M(\lambda)}{d\lambda^2} = \frac{3}{4} \int_0^s [\lambda \cdot f + (1 - \lambda) \cdot g]^{-\frac{5}{2}} \cdot (g - f)^2 \cdot d\sigma \quad (47)$$

From (a),(c)  $\frac{d^2M(\lambda)}{d\lambda^2} > 0$ , thus  $M(\lambda)$  is convex. ■

### Method of Martensson

In brief, this method suggests that a hard constraint of the form

$$G(x(s), s) \geq 0 \quad (48)$$

can be arbitrarily closely approximated by

$$g(x(s), u(s), s) = G^{(l)}(x(s), u(s), s) + \alpha_1 \cdot G^{(l-1)}(x(s), u(s), s) + \dots + \alpha_l \cdot G(x(s), s) \geq 0 \quad (49)$$

where

$$G^{(i)}(x(s), u(s), s) = \frac{dG^{(i-1)}(x(s), s)}{ds} = \frac{\partial G^{(i-1)}(x(s), s)}{\partial x} \cdot x' + \frac{\partial G^{(i-1)}(x(s), s)}{\partial s} \quad (50)$$

by choosing constants  $\alpha_i \in \mathfrak{R}$ ,  $i = 1, \dots, l$  to be the coefficients of a stable polynomial with real roots

$$\lambda^l + \alpha_l \cdot \lambda^{l-1} + \dots + \alpha_1 = 0. \quad (51)$$

The more stable is the polynomial the better is the approximation of the original constraints. Notice that because of the structure (5) of the state equations, function  $g(\cdot, \cdot, \cdot)$  is going to be of the form

$$g(x(s), u(s), s) = \mathcal{C}(x, s) + \mathcal{D}(x, s) \cdot u(s) \quad (52)$$

The order  $l$  is chosen such that in (50) a factor containing  $u$  explicitly exists. ■

#### Acknowledgment

This work has been supported by the NASA Center for Intelligent Robotic Systems for Space Exploration (CIRSSE) under the NASA Grant NAGW-1333. The first author has been partially supported by a fellowship of Alexander Onasis Foundation.

#### References

- [Can86] J. Canny. Collision detection for moving polyhedra. *IEEE Trans. Patt. Anal. Mach. Intell.*, pages 200–209, March 1986.
- [FH84] E. Freund and H. Hoyer. Collision avoidance for industrial robots with arbitrary motion. *Journal of Robotic Systems*, 1984.
- [FS90] K. Fujimura and H. Samet. Motion planning in a dynamic domain. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 324–330, 1990.
- [GE90] N. Griswold and J. Eem. Control of mobile robots in the presence of moving objects. *IEEE Trans. Robotics and Automation*, pages 263–268, April 1990.
- [GJ85] E. Gilbert and D. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, pages 21–30, March 1985.
- [K.K91] K. Kyriakopoulos. A supervisory control strategy for navigation of mobile robots in dynamic environments – ph.d thesis. Technical report, ECSE-RPI, February 1991.
- [KS90a] K. Kyriakopoulos and G. Saridis. Minimum distance estimation and collision prediction under uncertainty for on-line robotic motion planning. In *Proceedings of the 1990 IFAC World Congress*, August 1990.
- [KS90b] K. Kyriakopoulos and G. Saridis. On-line motion planning for mobile robots in non-stationary environments. In *Proceedings of 1990 IEEE International Symposium on Intelligent Control*, 1990.



- [KS91] K. Kyriakopoulos and G. Saridis. Collision avoidance of mobile robots in non-stationary environments. In *Proceedings of the 1991 International Conference on Robotics and Automation*, April 1991.
- [KZ84] K. Kant and S. Zucker. Trajectory planning problems, i: Determining velocity along a fixed path. In *Proceeding of the IEEE 8th Int. Conf. Pattern Recogn.*, pages 196–198, 1984.
- [KZ86] K. Kant and S. Zucker. Toward efficient trajectory planning: The path–velocity decomposition. *International Journal of Robotics Research*, pages 72–89, 1986.
- [KZ88] K. Kant and S. Zucker. Planning collision-free trajectories in time-varying environments: A two-level hierarchy. In *Proceeding of the 1988 IEEE International Conference on Robotics and Automation*, pages 1644–1649, April 1988.
- [Le89] Y. Liu and et.al. A practical algorithm for planning collision free coordinated motion of multiple mobile robots. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1427–1432, May 1989.
- [Mar73] K. Martensson. A constraining hyperplane technique for state variable constrained optimal control problems. *Journal of Dynamical Systems, Measurement and Control*, pages 380–389, 1973.
- [Pow83] M.J.D Powell. Variable metric methods for constraint optimization. *Mathematical Programming: The State of the Art*, pages 288–311, 1983.
- [RS85] J. Reif and M. Sharir. Motion planning in the presense of moving obstacles. Technical Report TR-06-85, Harvard University, Center for Research in Computing Technology, 1985.
- [SLG90] C. Shih, T. Lee, and W. Gruver. Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 331–337, 1990.
- [Tou86] P. Tournassoud. A strategy for obstacle avoidance and its applications to multi-robot systems. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 1224–1229, 1986.
- [WJ88] C. Wu and C. Jou. Design of a controlled spatial curve trajectory for robot manipulators. In *Proceedings of the 27th Conference on Decision and Control*, pages 161–166, December 1988.