Path-Planning with Collision Avoidance in Automotive Industry

Chantal Landry, Matthias Gerdts, René Henrion, Dietmar Hömberg

Path-Planning with Collision Avoidance in Automotive Industry

Chantal Landry[1], Matthias Gerdts[2], René Henrion[1], and Dietmar Hömberg[1]

[1] Weierstrass Institute, Mohrenstr. 39, 10117 Berlin, Germany
{chantal.landry,rene.henrion,dietmar.hoemberg}@wias-berlin.de
[2] Institute of Mathematics and Applied Computing, University of the Federal Armed Forces at Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
matthias.gerdts@unibw.de

Abstract. An optimal control problem to find the fastest collision-free trajectory of a robot is presented. The dynamics of the robot is governed by ordinary differential equations. The collision avoidance criterion is a consequence of Farkas's lemma and is included in the model as state constraints. Finally an active set strategy based on backface culling is added to the sequential quadratic programming which solves the optimal control problem.

Keywords: Optimal control, collision avoidance, backface culling, active set strategy

1 Collision Avoidance

In automotive industry robots have to work simultaneously on the same workpiece. The challenge is to find for each robot the fastest trajectory that avoids any collision with the surrounding obstacles and the other robots. We start with the establishment of the collision avoidance criterion.

For simplicity, we suppose that only one obstacle is present in the workspace. As in [7, 8] a collision detection can be obtained when the robot is approximated by a union of convex polyhedra. This union is called P and it given by

$$P = \bigcup_{i=1}^{n_P} P^{(i)}, \text{ with } P^{(i)} = \{y \in \mathbb{R}^3 | A^{(i)} y \le b^{(i)}\}$$

where n_P is the number of polyhedra in P. If p_i denotes the number of faces in $P^{(i)}$, then $A^{(i)} \in \mathbb{R}^{p_i \times 3}$ and $b^{(i)} \in \mathbb{R}^{p_i}$ for $i = 1, \ldots, n_P$.

Similarly, the obstacle is approximated by the following union of convex polyhedra, called Q

$$Q = \bigcup_{j=1}^{n_Q} Q^{(j)}, \text{ with } Q^{(j)} = \{y \in \mathbb{R}^3 | C^{(j)} y \le d^{(j)}\}$$

where n_Q is the number of polyhedra in Q. If q_j is the number of faces in $Q^{(j)}$, then $C^{(j)} \in \mathbb{R}^{q_j \times 3}$ and $d^{(j)} \in \mathbb{R}^{q_j}$ for $j = 1, \ldots, n_Q$. In the following, n_P, A, b

and i are always associated with the robot, and n_Q, C, d, and j are related to the obstacle. Furthermore, the robot will be identified with its approximation P and the obstacle with Q.

A first characterization of the collision-freeness between P and Q is given by

$$P^{(i)} \cap Q^{(j)} \ = \ \emptyset, \quad \forall i = 1, \ldots, n_P \text{ and } \forall j = 1, \ldots, n_Q.$$

The definition of the polyhedra $P^{(i)}$ and $Q^{(j)}$ implies that $P^{(i)}$ does not collide with $Q^{(j)}$ if and only if there does not exist any point $y^{(i,j)} \in \mathbb{R}^3$ satisfying

$$\begin{pmatrix} A^{(i)} \\ C^{(j)} \end{pmatrix} y^{(i,j)} \ \leq \ \begin{pmatrix} b^{(i)} \\ d^{(j)} \end{pmatrix}.$$

According to Farkas's lemma [1], this linear system does not have any solution if and only if there exists a vector $w^{(i,j)} \in \mathbb{R}^{p_i + q_j}$ such that

$$w^{(i,j)} \ \geq \ 0, \ \begin{pmatrix} A^{(i)} \\ C^{(j)} \end{pmatrix}^\top w^{(i,j)} \ = \ 0 \ \text{ and } \ \begin{pmatrix} b^{(i)} \\ d^{(j)} \end{pmatrix}^\top w^{(i,j)} < 0.$$

In conclusion, the pair of polyhedra $(P^{(i)}, Q^{(j)})$ is collision-free if and only if such a vector $w^{(i,j)}$ exists. This forms the collision avoidance characterization between a pair of polyhedra. Between the robot and the obstacle, the characterization reads as follows:

Proposition 1. *Two unions of convex polyhedra $P = \bigcup_{i=1}^{n_P} P^{(i)}$ and $Q = \bigcup_{j=1}^{n_Q} Q^{(j)}$ do not collide if and only if for each pair of polyhedra $(P^{(i)}, Q^{(j)})$, $i = 1, \ldots, n_P$, $j = 1, \ldots, n_Q$, there exists a vector $w^{(i,j)} \in \mathbb{R}^{p_i + q_j}$ such that*

$$w^{(i,j)} \ \geq \ 0, \ \begin{pmatrix} A^{(i)} \\ C^{(j)} \end{pmatrix}^\top w^{(i,j)} \ = \ 0 \ \text{ and } \ \begin{pmatrix} b^{(i)} \\ d^{(j)} \end{pmatrix}^\top w^{(i,j)} \ < \ 0.$$

2 Optimal Control Problem

To express the dynamics of the robot, we need to describe P differently from the previous section. As an industrial robot, P is composed by m links and is asked to move from its current position to a desired point [12]. Let $q = (q_1, \ldots, q_m)$ denote the vector of joint angles at the joints of the robot. The vector $v = (v_1, \ldots, v_m)$ contains the joint angle velocities and $u = (u_1, \ldots, u_m)$ describes the torques applied at the center of gravity of each link. The Lagrangian form of the dynamics of the robot depends on these three vectors as follows

$$q'(t) \ = \ v(t) \ \text{ and } \ \mathcal{M}(q(t)) v'(t) \ = \ \mathcal{G}(q(t), v(t)) + \mathcal{F}(q(t), u(t)), \qquad (1)$$

where $\mathcal{M}(q)$ is the symmetric and positive definite mass matrix, $\mathcal{G}(q, v)$ contains the generalized Coriolis forces and $\mathcal{F}(q, u)$ is the vector of applied joint torques and gravity forces [10, 12].

For the remainder of the paper, let us define the vector $x = (q, v) \in \mathbb{R}^{n_x}$ with $n_x := 2m$. With the definition of x and the non-singularity of the matrix $\mathcal{M}$, we can define the function $f : \mathbb{R}^{n_x} \times \mathbb{R}^m \to \mathbb{R}^{n_x}$ as follows

$$f(x, u) = \begin{pmatrix} v \\ \mathcal{M}^{-1}(q) \left(\mathcal{G}(q, v) + \mathcal{F}(q, u) \right) \end{pmatrix}.$$

The fastest trajectory of a robot is the solution of an optimal control problem where the system of ordinary differential equations (ODE) are given by (1), see [3, 6]. If an obstacle is present in the workspace, the collision-freeness is assured as soon as the vector $w^{(i,j)}$ of Proposition 1 is found at each time t and for all pairs of polyhedra. However, to be written as state constraints, the strict inequality in Proposition 1 has to be relaxed. Furthermore, since the robot moves, the matrices $A^{(i)}$ and the vectors $b^{(i)}$ evolves in time. Their evolution depends explicitly on $x(t)$. A complete formulation of $A^{(i)}(x(t))$ and $b^{(i)}(x(t))$ is given in [6].

Before writing down the model, let us define the index transformation $I = (i - 1)n_Q + j$. Hence, to each pair $(i, j) \in \{1, \ldots, n_P\} \times \{1, \ldots, n_Q\}$ there corresponds an index I in $\{1, \ldots, n_P n_Q\}$, and reciprocally. In the sequel, the index I is used instead of the pair (i, j). The variable $w^{(i,j)}$ is then numbered as w_I. Let us also define the functions $G_I : \mathbb{R}^{n_x} \to \mathbb{R}^{(p_i + q_j) \times 3}$ and $g_I : \mathbb{R}^{n_x} \to \mathbb{R}^{p_i + q_j}$ for $I = 1, \ldots, n_P n_Q$ as follows

$$G_I(x) = \begin{pmatrix} A^{(i)}(x) \\ C^{(j)} \end{pmatrix} \text{ and } g_I(x) = \begin{pmatrix} b^{(i)}(x) \\ d^{(j)} \end{pmatrix}.$$

G_I and g_I allow us to write Proposition 1 as a function of time. Finally let set $M := n_P n_Q$ the number of indices I and $n_I := p_i + q_j$ the size of w_I for $I = 1, \ldots, M$, and let t_f denote the travel time. Then, after transformation onto the fixed time interval $T := [0, 1]$ the optimal control problem reads as follows:

(OCP):   *minimize* $\varphi(x(0), x(1), t_f)$

with respect to the state variable $x \in W_{1,\infty}^{n_x}(T)$, the control variables $u \in L_\infty^m(T)$ and $w_I \in L_\infty^{n_I}(T)$, $I = 1, \ldots, M$, and $t_f \geq 0$, subject to:

- *ODE:* $x'(t) - t_f f(x(t), u(t)) = 0,$ *a.e. in T,*
- *state constraints:* $G_I(x(t))^\top w_I(t) = 0,$ $I = 1, \ldots, M,$ *a.e. in T,*   (2)

                        $g_I(x(t))^\top w_I(t) \leq -\varepsilon,$ $I = 1, \ldots, M,$ *a.e. in T,*   (3)

- *boundary conditions:* $\psi(x(0), x(1)) = 0,$
- *box constraints:* $w_I(t) \geq 0,$ $I = 1, \ldots, M,$ *a.e. in T,*

                        $u(t) \in \mathcal{U} := \{u \in \mathbb{R}^m \,|\, u_{min} \leq u \leq u_{\max}\}.$

where the function $\psi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{2n_x}$ is given as follows $\psi(x(t_0), x(t_f)) = (R(q(t_0)) - R_0, 0, R(q(t_f)) - R_f, 0)$ where $R(q)$ denotes the position of the barycenter of the last link of the robot and $R_0, R_f \in \mathbb{R}^m$ are given. The vectors u_{min} and u_{max} are also given. The relaxation parameter ε is positive and small. As usual $L_\infty^m(T)$ denotes the Banach space of essentially bounded functions mapping

from T into $\mathbb{R}^m$ and $W_{1,\infty}^{n_x}(T)$ denotes the Banach space of absolutely continuous functions with essentially bounded derivative that map from T into $\mathbb{R}^{n_x}$.

If multiple obstacles are present in the workspace, the *anti-collision constraints* (2)-(3) and the associated control variables w_I are defined for each obstacle.

Depending on the number M of anti-collision constraints, the problem is inherently sparse since the artificial control variables w_I, $I = 1, \ldots, M$, do not enter the dynamics, the boundary conditions, and the objective function of the problem, but only appear linearly in the anti-collision constraints with one-sided coupling through the state.

We attempt to solve the problem (OCP) numerically with a reduced discretization approach [4]. Let us consider the control grid $\mathbb{G}_N := \{t_k = kh \mid k = 0, 1, \ldots, N\}$, which, for simplicity, is chosen equidistantly with the fixed step-size $h = 1/N$. We use B-spline of second order to approximate the control variables:

$$u_h(t; u_0, \ldots, u_N) := \sum_{i=0}^{N} u_i\, B_i(t),$$

$$w_{I,h}(t; w_{I,0}, \ldots, w_{I,N}) := \sum_{i=0}^{N} w_{I,i}\, B_i(t), \quad I = 1, \ldots, M$$

where $(u_0, \ldots, u_N)^\top \in \mathbb{R}^{m(N+1)}$ and $(w_{I,0}, \ldots, w_{I,N})^\top \in \mathbb{R}^{n_I(N+1)}$ are the vector of de Boor points, and B_i, $i = 0, \ldots, N$, denote elementary B-splines. As the elementary B-splines sum up to one for all $t \in T$, the box constraints $u_h(t) \in \mathcal{U}$ are satisfied, if $u_i \in \mathcal{U}$, $i = 0, \ldots, N$. The choice of B-splines is convenient as it is easy to create approximations with prescribed smoothness properties and, even more important, the elementary B-splines B_i have a local support only.

We solve the differential equations for the initial value x_0 and a given t_f by the classical explicit Runge-Kutta method of order 4. The state approximations at the grid points t_k, $k = 0, \ldots, N$ depend on the vector $z := (x_0, u_0, \ldots, u_N, t_f)^\top \in \mathbb{R}^{n_z}$ with $n_z = n_x + (N+1)m + 1$.

Let us define $J(z) := \varphi(x_0, x_N(z), t_f)$, $h(z) := \psi(x_0, x_N(z))$, as well as $\bar{G}_{I,k}(z) := G_I(x_k(z))$ and $\bar{g}_{I,k}(z) := g_I(x_k(z))$ for $I = 1, \ldots, M$, $k = 0, \ldots, N$. With these new notations the discretized form of (OCP) can be formulated as follows

(DOCP):   *Minimize $J(z)$ with respect to $z \in \mathbb{R}^{n_z}$ and*
$$w = (w_{1,0}, \ldots, w_{1,N}, \ldots, w_{M,0}, \ldots, w_{M,N})^\top \in \mathbb{R}^{(N+1)\sum_{I=1}^{M} n_I}$$

subject to: $h(z) = 0,$
$$w_{I,k} \geq 0, \qquad I = 1, \ldots, M,\ k = 0, \ldots, N,$$
$$\bar{G}_{I,k}(z)^\top w_{I,k} = 0, \qquad I = 1, \ldots, M,\ k = 0, \ldots, N,$$
$$\bar{g}_{I,k}(z)^\top w_{I,k} \leq -\varepsilon, \qquad I = 1, \ldots, M,\ k = 0, \ldots, N,$$
$$z \in \mathcal{Z} := \{z \in \mathbb{R}^{n_z} \mid z_\ell \leq z \leq z_u\}.$$

Herein, $z_\ell \leq z_u$ define box constraints for z, where the settings $\pm\infty$ are permitted if a component of z is not restricted from above or below.

The above nonlinear optimization problem is solved by a sequential quadratic programming (SQP) method [5, 9]. As in [13] we use an Armijo type line-search procedure for the augmented Lagrangian function in our implementation. However, (DOCP) contains a lot of constraints: at each time step t_k, $k = 0, \ldots, N$, and for every pair of polyhedra $(P^{(i)}, Q^{(j)})$, four anti-collision constraints are defined (compare (2)-(3)). To overcome this difficulty, we add an active set strategy based on the following observation: the anti-collision constraints are superfluous when the robot is far from the obstacle or moves in the opposite direction. The establishment of the active set strategy is the purpose of the next section.

3 Backface Culling Active Set Strategy

Backface culling comes from computer graphics and consists of working only with visible objects, see [14]. We apply here the same concept to define our active set strategy and develop four criteria to determine which objects are visible.

In this section P, resp. Q, is a polyhedron belonging to the approximation of the robot, resp. obstacle. The first criterion is similar to the *broad phase* in [2, 11] and consists of checking if P is far from Q. If this is the case, no collision can occur and the anti-collision constraints are superfluous.

The distance between P and Q is roughly computed by defining the axis-aligned bounding box of each polyhedron. Let (x_i^P, y_i^P, z_i^P), $i = 1, \ldots, s^P$, denote the vertices of P where s^P is the number of vertices, and define

$$x_m^P = \min_{i=1,\ldots,s^P} x_i^P, \quad y_m^P = \min_{i=1,\ldots,s^P} y_i^P, \quad z_m^P = \min_{i=1,\ldots,s^P} z_i^P,$$
$$x_M^P = \max_{i=1,\ldots,s^P} x_i^P, \quad y_M^P = \max_{i=1,\ldots,s^P} y_i^P, \quad z_M^P = \max_{i=1,\ldots,s^P} z_i^P.$$

Then, the tuple $(x_m^P, y_m^P, z_m^P, x_M^P, y_M^P, z_M^P)$ represents the smallest axis-aligned bounding box around P. Similarly the tuple $(x_m^Q, y_m^Q, z_m^Q, x_M^Q, y_M^Q, z_M^Q)$ denotes the smallest bounding box of Q. Let $\delta > 0$ and define slightly bigger boxes:

$$\mathcal{B}^P = [x_m^P - \delta, \, x_M^P + \delta] \times [y_m^P - \delta, \, y_M^P + \delta] \times [z_m^P - \delta, \, z_M^P + \delta]$$
$$\mathcal{B}^Q = [x_m^Q - \delta, \, x_M^Q + \delta] \times [y_m^Q - \delta, \, y_M^Q + \delta] \times [z_m^Q - \delta, \, z_M^Q + \delta].$$

With these definitions the first criterion reads

Criterion 1 *If $\mathcal{B}^P$ and $\mathcal{B}^Q$ are separated, then P is far from Q and the associated anti-collision constraints are superfluous.*

The vertices of P evolve in time since they belong to the robot. Hence the box $\mathcal{B}^P$ has to be determined at each grid point t_k, $k = 0, \ldots, N$. Because the obstacle does not move, the box $\mathcal{B}^Q$ is computed only once.

Let us assume now that Q is close enough to P and consider the situation depicted in Figure 2: P is moving downwards, v_c indicates the velocity of the

center of gravity and $\tilde{Q}$ is generated by the faces e_1 and e_2 of Q. According to Proposition 1 $\tilde{Q}$ does not collide with P if and only if

$$\exists \tilde{w} > 0, \text{ such that } \begin{pmatrix} A \\ C_{1,2} \end{pmatrix}^\top \tilde{w} = 0 \text{ and } \begin{pmatrix} b \\ d_{1,2} \end{pmatrix}^\top \tilde{w} < 0, \qquad (4)$$

where $C_{1,2}$ is the matrix composed of the first two rows of C and $d_{1,2}$ is the vector composed of the first two components of d.

(a)   (b)
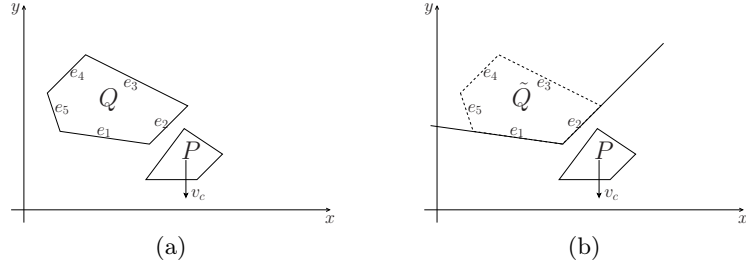
Fig. 1. (a) The polyhedron P is moving downwards. The faces of Q are denoted by $e_1, \ldots, e_5$. (b) The set $\tilde{Q}$ is generated by the faces of Q visible to P.

Suppose now that $\tilde{w}$ exists. By setting $w = (\tilde{w}, 0, 0, 0)$, we obtain:

$$\begin{pmatrix} A \\ C \end{pmatrix}^\top w = 0 \text{ and } \begin{pmatrix} b \\ d \end{pmatrix}^\top w < 0.$$

Then, Proposition 1 implies that P and Q do not collide. In summary, if no collision occurs between $\tilde{Q}$ and P, then Q and P do not collide. The dimension of $\tilde{w}$ is always smaller than that of w, because the polyhedra are supposed to be compact. Consequently, the problem of finding $\tilde{w}$ is always smaller and there is an advantage in replacing the anti-collision constraints by (4). In (4) only the faces visible to P are taken into consideration. The next criteria concern the determination of the visible faces of Q relative to P.

The faces of Q which are located behind P, are invisible to P. P is always looking towards its velocity, v_c. Hence, all objects located in the lower halfspace $\mathcal{H}$ generated by v_c and S_R, the vertex of P located furthest in the opposite direction of v_c, are behind P. Then, the second criterion reads

Criterion 2 *A face e of Q is invisible to P if $e \subset \mathcal{H} = \{y \in \mathbb{R}^n | v_c^\top (y - S_R) < 0\}$.*

An example is given in Figure 2-(a) where the faces e_3 and e_4 satisfy Criterion 2. The case where all faces are located behind P means that P is moving in the opposite direction to Q. In this situation no collision can occur and we have:

Criterion 3 *If all faces of Q are invisible to P according to Criterion 2, then the anti-collision constraints are superfluous.*

Remark 1. If the velocity v_c is zero, Criteria 2 and 3 are not applied.

Not all remaining faces of Q are visible to P. Some of them can be hidden by other faces of Q. This is the case of the face e_5 in Figure 2-(b). The vertex S, which is the closest point of P to e_5, cannot see e_5 because e_5 is hidden by e_1. The face e_5 will be visible to S as soon as S is no more located in the halfspace $\mathcal{H}_5 = \{y \in \mathbb{R}^n \,|\, C_5\,y < d_5\}$. This yields the last criterion

Criterion 4 *The face e_i of Q is invisible to P if $P \subset \mathcal{H}_i = \{y \in \mathbb{R}^n | C_i y < d_i\}$ with C_i, the ith line of C, and d_i, the ith component of d, in the definition of Q.*
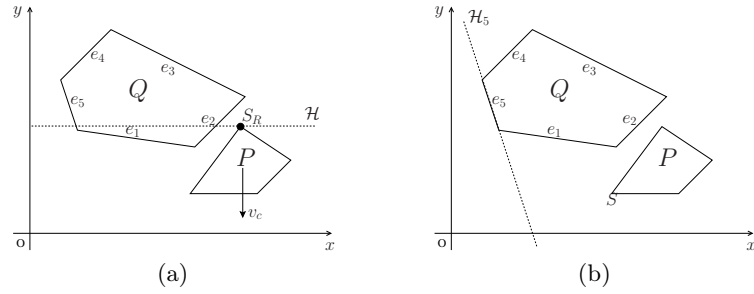
Fig. 2. (a) The faces e_3 and e_4 of Q are located behind P. (b) The face e_5 of Q is invisible to P.

A limit case exists with Criterion 4 when P is included in Q. In that case all faces of Q are invisible to P according to Criterion 4. But in fact all these faces must be considered in the anti-collision constraints. Hence, Criterion 4 must not be applied in this particular case.

Criterion 4 can also be applied to detect which faces of P are visible to Q. Then the anti-collision constraints defined for the pair (P, Q) can be reduced as it was done in (4).

In this section criteria to determine the visible faces of Q were developed, provided Q is visible. All criteria were depending on the position of P which is given by the state variable q. In the next section we show how the backface culling is included in the SQP algorithm to solve (DOCP).

4   Algorithm and numerical examples

Let us recall the index transformation that associates to each pair (i, j) the new index I via the formula: $I = (i - 1)n_Q + j$ and define the set of indices

$$\mathcal{K} := \{(I, k) \,|\, \text{the polyhedron } Q^{(j)} \text{ is visible to } P^{(i)} \text{ at } t_k\}.$$

$\mathcal{K}$ is determined by applying Criteria 1, 2 and 3. Let us also recall that w_I belongs to $\mathbb{R}^{p_i + q_j}$. The first p_i components of w_I are associated to the faces of

$P^{(i)}$ and the next q_j components are related to the faces of $Q^{(j)}$. Then, let us define the following set of indices for each $(I, k) \in \mathcal{K}$

$$\mathcal{J}_{I,k} := \{c \in \{1, \ldots, p_i\} \,|\, \text{the face } c \text{ of } P^{(i)} \text{ is invisible to } Q^{(j)} \text{ at } t_k\} \cup$$

$$\{c \in \{p_i + 1, \ldots, p_i + q_j\} \,|\, \text{the face } c - p_i \text{ of } Q^{(j)} \text{ is invisible to } P^{(i)} \text{ at } t_k\}.$$

This set contains the index of the faces of the pair $(P^{(i)}, Q^{(j)})$ which are invisible at t_k. The invisibility of a face is determined using Criteria 2 and 4.

Backface culling involves considering the anti-collision constraints whose pair of indices (I, k) belongs to $\mathcal{K}$ and write these constraints according to (4). The algorithm to solve $(DOCP)$ is the SQP method presented below in which the backface culling is added as an active set strategy. This means that at each iteration we update the set $\mathcal{K}$ and then build the quadratic problem by considering only the constraints whose pair of indices belongs to $\mathcal{K}$.

Backface Culling Active Set Strategy

(0) Choose $\varepsilon > 0$, $z^{(0)} \in \mathcal{Z}$ and $w^{(0)} \geq 0$.
   Determine the sets of indices $\mathcal{K}^{(0)}$ and $\mathcal{J}_{I,k}^{(0)}$ for all $(I, k) \in \mathcal{K}^{(0)}$.
   Set $B_0 := I$, the identity matrix and $\ell := 0$.

(1) If $(z^{(\ell)}, w^{(\ell)})$ is a KKT point of the optimization problem, STOP.

(2) Compute a KKT point of the following linear-quadratic optimization problem:
   $$\text{Minimize} \quad \tfrac{1}{2} d^\top B_\ell d + J'(z^{(\ell)}) d_z$$

   with respect to $d = (d_z, d_{w_{I,k}})$, $(I, k) \in \mathcal{K}^{(\ell)}$, subject to the constraints

$$h(z^{(\ell)}) + h'(z^{(\ell)}) d_z = 0,$$
$$w_{I,k}^{(\ell)} + d_{w_{I,k}} \geq 0, \qquad (I, k) \in \mathcal{K}^{(\ell)},$$
$$\bar{G}_{I,k}(z^{(\ell)})^\top w_{I,k}^{(\ell)} + \bar{G}_{I,k}(z^{(\ell)})^\top d_{w_{I,k}}$$
$$+ \bar{G}'_{I,k}(z^{(\ell)})^\top (w_{I,k}^{(\ell)}, d_z) = 0, \qquad (I, k) \in \mathcal{K}^{(\ell)},$$
$$\bar{g}_{I,k}(z^{(\ell)})^\top w_{I,k}^{(\ell)} + \bar{g}_{I,k}(z^{(\ell)})^\top d_{w_{I,k}}$$
$$+ \bar{g}'_{I,k}(z^{(\ell)})^\top (w_{I,k}^{(\ell)}, d_z) \leq -\varepsilon, \qquad (I, k) \in \mathcal{K}^{(\ell)},$$
$$z^{(\ell)} + d_z \in \mathcal{Z},$$
$$d_{w_{I,k},c} = 0, \qquad c \in \mathcal{J}_{I,k}^{(\ell)}, (I, k) \in \mathcal{K}^{(\ell)}.$$

   Note: The constraints $d_{w_{I,k},c} = 0$ are only included for notational simplicity. In practice these variables are actually eliminated.

(3) Set
   $$z^{(\ell+1)} := z^{(\ell)} + d_z^{(\ell)}, \quad w_{I,k}^{(\ell+1)} := w_{I,k}^{(\ell)} + d_{w_{I,k}}^{(\ell)}, \ (I, k) \in \mathcal{K}^{(\ell)}.$$

(4) Update the sets of indices $\mathcal{K}^{(\ell+1)}$ and $\mathcal{J}_{I,k}^{(\ell+1)}$ for $(I, k) \in \mathcal{K}^{(\ell+1)}$ according to Criteria 1 to 4 which depend on $z^{(\ell+1)}$. Update $B_{\ell+1}$ according to BFGS update formulas, set $\ell := \ell + 1$ and go to (1).

At the step (4), if a pair of indices (I, k) newly appears in $\mathcal{K}^{(\ell+1)}$ (i.e. if $(I, k) \in \mathcal{K}^{(\ell+1)} \setminus \mathcal{K}^{(\ell)}$), then the variable $w_{I,k}^{(\ell+1)}$ must be initialized. We choose to take $w_{I,k}^{(\ell+1)}$ as the solution of

$$\min_w \bar{g}_{I,k}(z^{(\ell+1)})^\top w \text{ such that } \bar{G}_{I,k}(z^{(\ell+1)})^\top w = 0,$$
$$w_c > 0, \text{ if } c \notin \mathcal{J}_{I,k}^{(\ell+1)}, \text{ and } w_c = 0, \text{ if } c \in \mathcal{J}_{I,k}^{(\ell+1)}.$$

Thereby, we are assured to satisfy the state constraints as close as possible.

In our first example we consider an obstacle and a robot composed by a socket and 3 links. A load is fixed at the end of the last link. A complete description of the example is given in [6]. In this example, the collision avoidance needs to be applied only between the load and the obstacle. The obstacle is always close to the load. Consequently, the number of state contraints is not reduced with the backface culling. For this example, we take 21 control grid points and $\varepsilon = 10^{-5}$. In Figure 3 the visible faces of the obstacle are in white. We can observe that only 3 faces of the obstacle are visible. The computational time is $52\,s$. If we do not use the backface culling, the computational time is equal to $3\,min\,44$. So, with about half of the unknowns, the code runs about four times faster.

at t_1 at t_4 at t_9 at t_{14} at t_{17}
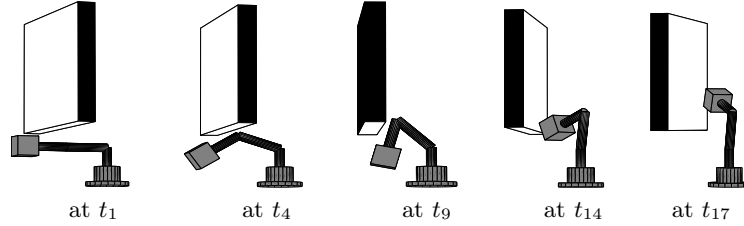
Fig. 3. Snapshots of the motion of the robot avoiding an obstacle. The visible faces of the obstacle are in white.

The second example is in 2 dimensions and uses all criteria of the backface culling. The robot P is a square and 4 obstacles, Q_1 to Q_4, are present in the workspace. We take for this example 42 control grid points and $\varepsilon = 10^{-2}$. Snapshots of the motion of the robot, which must reach the point F, are given in Figure 4. The visible obstacles are in white and their visible faces in gray. The computational time is equal to $36\,min\,50$ when no backface culling is used. With the backface culling strategy the CPU time is $27\,s$. Hence, Criteria 1 to 4 induce a large decrease in the computational time.

at t_1 at t_{14} at t_{33} at t_{39}
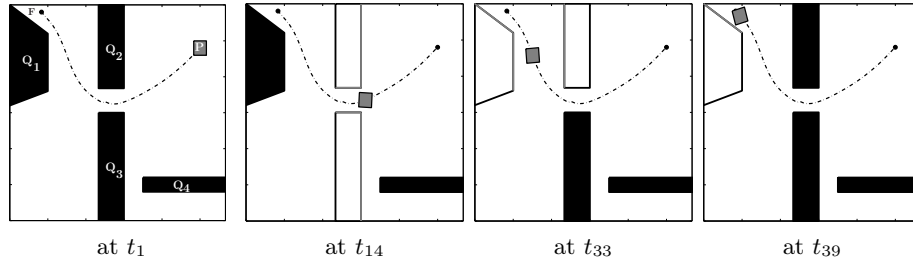
Fig. 4. Snapshots of the motion of the robot P moving to F and avoiding four obstacles. The visible obstacles are in white and their visible faces in gray.

References

1. Berkovitz, L.D.: Convexity and Optimization in $\mathbb{R}^n$. John Wiley & Sons, New York, (2001)
2. Cohen, J.D., Lin, M.C., Manocha, D., Ponamgi, M.K.: I-collide: An Interactive and Exact Collision Detection System for Large-Scaled Environments. In: Symposium on Interactive 3D Graphics, pp. 189-196. ACM Siggraph (1995)
3. Diehl, M., Bock, H.G., Diedam, H., Wieber, P.B.: Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In: Fast Motions in Biomechanics and Robotics, pp. 65–94. Springer (2005)
4. Gerdts, M.: Direct Shooting Method for the Numerical Solution of Higher-Index DAE Optimal Control Problems. Journal of Optimization Theory and Applications, 117, 267–294 (2003)
5. Gerdts, M.: Optimal Control of Ordinary Differential Equations and Differential-Algebraic Equations. Fakultät für Mathematik und Physik, Universität Bayreuth, Bayreuth (2006)
6. Gerdts, M., Henrion, R., Hömberg, D., Landry, C.: Path Planning and Collision Avoidance for Robots. Numer. Algebra Control Optim., 3, 437–463 (2012)
7. Gilbert, E.G., Hong, S.M.: A New Algorithm for Detecting the Collision of Moving Objects. In: IEEE Proc. Int. Conf. on Robotics and Automation, 1, pp. 8–14 (1989)
8. Gilbert, E.G., Johnson, D.W.: Distance Functions and their Application to Robot Path Planning in the Presence of Obstacles. IEEE Journal of Robotics and Automation, 1, 21–30 (1985)
9. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM Rev. 47, 99–131 (2005)
10. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
11. Mirtich, B.: V-Clip: Fast and Robust Polyhedral Collision Detection. Mitsubishi Electronics Research Laboratory, (1997)
12. Murray, R. M., Sastry, S. S., Li, Z.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Inc., Boca Raton (1994)
13. Schittkowski, K.: On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangean Line Search Function. In: Mathematische Operationsforschung und Statistik, Series Optimization 14, 197–216 (1983)
14. Vaněček Jr., G.: Back-Face Culling Applied to Collision Detection of Polyhedra. Journal of Visualization and Computer Animation, 5, 55–63 (1994)