Path planning and collision avoidance for robots

Dedicated to Prof. Dr. Helmut Maurer on the occasion of his 65th birthday

Matthias Gerdts\textsuperscript{1}, René Henrion\textsuperscript{2}, Dietmar Hömberg\textsuperscript{2}, Chantal Landry\textsuperscript{2}

submitted: October 27, 2011

\textsuperscript{1} Institute of Mathematics and Applied Computing (LRT)
University of the Federal Armed Forces at Munich
Werner-Heisenberg-Weg 39
85577 Neubiberg
Germany
E-Mail: matthias.gerdts@unibw.de

\textsuperscript{2} Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: rene.henrion@wias-berlin.de
dietmar.hoemberg@wias-berlin.de
chantal.landry@wias-berlin.de

No. 1658
Berlin 2011

\textsuperscript{2010 Mathematics Subject Classification.} 49J15, 49M25, 49N90, 90C30.

\textit{Key words and phrases.} Optimal control, collision avoidance, cooperative robots, backface culling, active set strategy.

This work has been supported by the DFG Research Center \textsc{Matheon} – Mathematics for key technologies.
Abstract. An optimal control problem to find the fastest collision-free trajectory of a robot surrounded by obstacles is presented. The collision avoidance is based on linear programming arguments and expressed as state constraints. The optimal control problem is solved with a sequential programming method. In order to decrease the number of unknowns and constraints a backface culling active set strategy is added to the resolution technique.

1. Introduction

In this paper, we are interested in finding the time optimal trajectory of a robot. The manipulator must move from an initial position to a given final state without colliding with surrounding obstacles.

Industrial robots are an important part of the competitiveness of nowadays industries. They are asked to accomplish their task in a workspace containing obstacles as fast as possible or with a minimized energy consumption. The search of their optimal collision-free trajectory is an old and still topical question. The first significant paper dates from the seventies. Kahn and Roth had the idea to compute the optimal trajectory by means of control theory, see [21]. The research really started at the beginning of the eighties. See [5] for a good review. However, because of the high complexity of the problem the obstacles could not be taken into account in the model. The detection of collision was a different topic. In 1985, Gilbert and Johnson [16] have for the first time coupled these two problems. They worked on a simple 2 dimensional example. The collision avoidance was introduced as a state constraint in the optimal control problem and based on the distance between two objects. Since then, other authors developed similar models. Dubowsky, Norris and Shiller [7], Bobrow [3], and LaValle [22] also worked with the distance function. However, they were working along a specified path. In [15] Gilbert and Hong developed a model for a 3 dimensional manipulator. In this article, we will also express the collision avoidance as state constraints in the time optimal control problem. However, our constraints are not based on the distance function, which is non-differentiable in general, but as a consequence of the Farkas’s lemma. The use of Farkas lemma yields an algebraic formulation of the collision avoidance whose derivative is simple to obtain in comparison to the distance function.

Let us consider a robot composed by $m$ links connected to each other with revolute joints. The joint angles defined at the joints of the robot are stored in the vector $q$. The vector $v$ contains the joint angle velocities and $u$ represents the torques at the center of gravity of the robot links. As cited in [6] the use of optimal control problems is nowadays a natural technique when we want to find the best path of such a robot. The formulation of the optimal control problem (without collision avoidance) is given by:

$$\min \varphi(q(t_0), v(t_0), q(t_f), v(t_f), t_f)$$

subject to

- Ordinary differential equations

$$q' = v,$$

$$M(q)v' = G(q, v) + F(q, u),$$

(1)
• boundary conditions
  \[ q(0) = q_0, \ v(0) = 0, \ q(t_f) = q_f \text{ and } v(t_f) = 0, \]

• and box constraints
  \[ u_{\text{min}} \leq u \leq u_{\text{max}}. \]

In (1) \( M \) denotes the symmetric and positive definite mass matrix, the vector \( G \) is the generalized Coriolis forces and \( F \) denotes the vector of applied joint torques and gravity forces. Equation (1) is the Lagrangian form of the dynamics of a robot. The vectors \( q_0 \) and \( q_f \) are the given initial and terminal states. Details on the dynamics are given in Appendix A.

Let us define the vector

\[ x = \begin{pmatrix} q \\ v \end{pmatrix} \in \mathbb{R}^{n_x}, \text{ with } n_x := 2m. \]

This new vector contains the state variables of the optimal control problem. Let \( n_u \) denote the dimension of \( u \). In our case we have \( n_u = m \). With the definition of \( x \) and the non-singularity of the matrix \( M \), we can define the function \( f : \mathbb{R}^{n_x} \times \mathbb{R}^n \to \mathbb{R}^{n_x} \) as follows

\[ f(x, u) = \begin{pmatrix} v \\ M^{-1} (G(q, v) + F(q, u)) \end{pmatrix}. \]

The above optimal control problem can be then transformed into

\[ \min \ \varphi(x(t_0), x(t_f), t_f) \]

subject to

\[ x'(t) - f(x(t), u(t)) = 0, \quad a.e. \in [t_0, t_f], \]

\[ \psi(x(t_0), x(t_f)) = 0, \]

\[ u(t) \in \mathcal{U} := \{ u \in \mathbb{R}^{n_u} | u_{\text{min}} \leq u \leq u_{\text{max}} \}, \]

where the function \( \psi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{2n_x} \) contains the boundary conditions as follows

\[ \psi(x(t_0), x(t_f)) = \begin{pmatrix} q(t_0) - q_0 \\ v(t_0) \\ q(t_f) - q_f \\ v(t_f) \end{pmatrix}. \]

Diehl et al. [6] show that the direct approach is the best approach to solve such a system. We will follow this technique of “first discretize, then optimize” and use a sequential quadratic programming method to solve the resulting nonlinear problem.

A common technique to detect the collision between the robot and an obstacle is to describe these objects as a union of convex polyhedra. However, Chang and Kavraki pointed out in [20] that such a technique is not applicable in real-world problem. Indeed, real robots can contain complex geometries and the number of convex polyhedra to approach such geometries can explode. Pérez-Francisco et al. suggest in [23] to avoid to work with convex polyhedra and propose a method which can work directly with nonconvex and curved objects. We still use the convex division, but add a back-face culling strategy to avoid the explosion of the costs. This strategy consists of applying the collision avoidance only between polyhedra which are visible to each other. So, only a small number of convex polyhedra are needed to ensure the collision avoidance. Our technique is then still relevant for practical cases.
This article is divided as follows. Our collision avoidance criterion is introduced in Section 2. Section 3 contains the final form of the time optimal control problem where the collision avoidance is expressed as state constraints. The direct discretization of the optimal control problem is presented in Section 4. Section 5 points out the large size of the discretized optimal control problem and proposes the backface culling to decrease the number of unknowns and state constraints. At the end of this section the SQP algorithm coupled with the backface culling strategy is sketched. Finally, numerical results for a 3-link robot are given in Section 6.

2. Collision Avoidance

For simplicity, we suppose that only one obstacle is in the workspace. In the introduction, the robot is described by the variables $q$ and $v$. To establish the collision avoidance criterion, a different description is used. As in [15, 16] the robot is approximated by a union of convex polyhedra. This union is called $P$ and is given by

$$P = \bigcup_{i=1}^{n_p} P^{(i)}, \text{ with } P^{(i)} = \{x \in \mathbb{R}^3 | A^{(i)}x \leq b^{(i)}\}$$

where $n_p$ is the number of polyhedra in $P$ and for $i = 1, \ldots, n_p$, $A^{(i)} \in \mathbb{R}^{p_i \times 3}$, $b^{(i)} \in \mathbb{R}^{p_i}$ and $p_i$ is the number of faces in $P^{(i)}$. Every inequality in (2) describes a face of the polyhedron. The exponent "$(i)$" refers to the data defined for the $i$th polyhedron in $P$.

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

$$Q = \bigcup_{j=1}^{n_q} Q^{(j)}, \text{ with } Q^{(j)} = \{x \in \mathbb{R}^3 | C^{(j)}x \leq d^{(j)}\}$$

where $n_q$ is the number of polyhedra in $Q$ and for $j = 1, \ldots, n_q$, $C^{(j)} \in \mathbb{R}^{q_j \times 3}$, $d^{(j)} \in \mathbb{R}^{q_j}$ and $q_j$ is the number of faces in $Q^{(j)}$. The exponent "$(j)$" refers to the data defined for the $j$th polyhedron in $Q$.

In the following, the letters $n_p$, $A$, $b$ and $i$ are associated to the robot, and the letters $n_q$, $C$, $d$, and $j$ are related to the obstacle. Furthermore, the robot will be identified with its approximation $P$ and the obstacle with $Q$. A scheme is given in Figure 1 where $P$ is composed by five polyhedra, and $Q$ by three polyhedra.

**Figure 1.** Approximation of the robot and the obstacle by a union of convex polyhedra.

Using the approximations $P$ and $Q$, the robot and the obstacle do not collide if and only if

$$P^{(i)} \cap Q^{(j)} = \emptyset, \quad \forall i = 1, \ldots, n_p \text{ and } \forall j = 1, \ldots, n_q.$$
The definition of the polyhedra $P^{(i)}$ and $Q^{(j)}$ implies that the above relation is equivalent to the in-solvability of the following linear system

\[
\begin{pmatrix}
A^{(i)} \\
C^{(j)}
\end{pmatrix} x \leq \begin{pmatrix}
b^{(i)} \\
d^{(j)}
\end{pmatrix}, \quad \forall i = 1, \ldots, n_p \text{ and } \forall j = 1, \ldots, n_q.
\]

According to Farkas’s lemma [1], the system (3) has no solution if and only if there exists a vector $w^{(i,j)}$ such that

\[
w^{(i,j)} \geq 0, \quad \left( A^{(i)} C^{(j)} \right)^T w^{(i,j)} = 0 \text{ and } \left( b^{(i)} d^{(j)} \right)^T w^{(i,j)} < 0.
\]

In conclusion, the pair of polyhedra $(P^{(i)}, Q^{(j)})$ do not collide if and only if such a vector $w^{(i,j)}$ exists. This forms the collision avoidance criterion between a pair of polyhedra. Between the robot and the obstacle, the criterion reads

**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 $w^{(i,j)}$ such that

\[
w^{(i,j)} \geq 0, \quad \left( A^{(i)} C^{(j)} \right)^T w^{(i,j)} = 0 \text{ and } \left( b^{(i)} d^{(j)} \right)^T w^{(i,j)} < 0.
\]

Proposition 1 is established for motionless bodies. In practice the robot $P$ moves in the workspace. The approximation (2) then becomes

\[
P(t) = \bigcup_{i=1}^{n_p} P^{(i)}(t), \quad \text{with } P^{(i)}(t) = \{ x \in \mathbb{R}^3 | A^{(i)}(t)x \leq b^{(i)}(t) \}.
\]

The time dependency occurs in the matrix $A^{(i)}$ and the vector $b^{(i)}$. For notational simplicity, the definition of $P^{(i)}$ at $t = 0$ is just written as $P^{(i)}(0) = \{ x \in \mathbb{R}^3 | A^{(i)}x \leq b^{(i)} \}$. A motion of $P^{(i)}$ is the composition of a rotation with a translation. Mathematically, the motion takes the following form

\[
P^{(i)}(t) = S^{(i)}(t)P^{(i)}(0) + r^{(i)}(t),
\]

where $S^{(i)}(t)$ is the orthogonal matrix which describes the rotational motion and $r^{(i)}(t)$ is the translational motion vector. Hence, a point $\tilde{x}$ belongs to $P^{(i)}(t)$ if and only if $x \in P^{(i)}(0)$ exists such that

\[
\tilde{x} = S^{(i)}(t)x + r^{(i)}(t).
\]

The matrix $A^{(i)}(t)$ and the vector $b^{(i)}(t)$ depend on $S^{(i)}(t)$, $r^{(i)}(t)$ and the definition of $P^{(i)}$ at $t = 0$ in the following manner

**Theorem 2.1.** For all $t > 0$, we have

\[
A^{(i)}(t) = A^{(i)} S^{(i)}(t)^T \text{ and } b^{(i)}(t) = b^{(i)} + A^{(i)} S^{(i)}(t)^T r^{(i)}(t).
\]

**Proof.** Let us recall that $P^{(i)}(t)$ is defined by

\[
P^{(i)}(t) = \{ y \in \mathbb{R}^3 | A^{(i)}(t)y \leq b^{(i)}(t) \} = \{ y \in \mathbb{R}^3 | \exists x \in P^{(i)}(0) \text{ such that } y = S^{(i)}(t)x + r^{(i)}(t) \}.
\]
Let $\mathcal{T}$ denote the set

$$\mathcal{T} = \{ y \in \mathbb{R}^3 \mid A^{(i)} S^{(i)}(t)^T y \leq b^{(i)} + A^{(i)} S^{(i)}(t)^T r^{(i)}(t) \}.$$ 

The proof of the theorem consists of establishing that $P^{(i)}(t) = \mathcal{T}$. We start with the first inclusion $P^{(i)}(t) \subset \mathcal{T}$ and consider a point $\tilde{x}$ in $P^{(i)}(t)$. By definition of $P^{(i)}(t)$, there exists a point $x$ in $P^{(i)}(0)$ such that

$$\tilde{x} = S^{(i)}(t)x + r^{(i)}(t).$$

Let us multiply this equation by $A^{(i)} S^{(i)}(t)^T$. The orthogonality of $S^{(i)}$ yields

$$A^{(i)} S^{(i)}(t)^T \tilde{x} = A^{(i)} x + A^{(i)} S^{(i)}(t)^T r^{(i)}(t).$$

Because $x \in P^{(i)}(0)$, we have

$$A^{(i)} S^{(i)}(t)^T \tilde{x} \leq b^{(i)} + A^{(i)} S^{(i)}(t)^T r^{(i)}(t),$$

which means that $\tilde{x} \in \mathcal{T}$.

For the inclusion $\mathcal{T} \subset P^{(i)}(t)$, let us consider the point $x$ given by

$$x = S^{(i)}(t)^T (\tilde{x} - r^{(i)}(t))$$

and multiply this relation by $A^{(i)}$

$$A^{(i)} x = A^{(i)} S^{(i)}(t)^T \tilde{x} - A^{(i)} S^{(i)}(t)^T r^{(i)}(t).$$

Yet, $\tilde{x} \in \mathcal{T}$ implies

$$A^{(i)} x = A^{(i)} S^{(i)}(t)^T \tilde{x} - A^{(i)} S^{(i)}(t)^T r^{(i)}(t) \leq b^{(i)}.$$ 

It follows that the point $x$ belongs to $P^{(i)}(0)$. Consequently, $\tilde{x} \in P^{(i)}(t)$. \qed

Applying Proposition 1 at a fixed time $t$ with Theorem 2.1 gives the collision avoidance criterion for a moving robot.

**Proposition 2.** The robot $P(t)$ and the obstacle $Q$ do not collide if and only if for each pair of polyhedra $(P^{(i)}(t), Q^{(j)})$, $i = 1, \ldots, n_p$, $j = 1, \ldots, n_q$, there exists a vector $w^{(i,j)}(t) \in \mathbb{R}^{n_p+n_q}$ such that $w^{(i,j)}(t) \geq 0$,

$$\left( \begin{array}{c} A^{(i)} S^{(i)}(t)^T \\ C^{(j)} \end{array} \right)^T w^{(i,j)}(t) = 0$$

and

$$\left( \begin{array}{c} b^{(i)} + A^{(i)} S^{(i)}(t)^T r^{(i)}(t) \\ d^{(j)} \end{array} \right)^T w^{(i,j)}(t) \leq 0.$$

Hence, if a vector $w^{(i,j)}$ exists for each pair of polyhedra and for all $t$, then the robot does not collide with the obstacle. To obtain an optimal collision-free trajectory for the robot, all we have to do is to find such a vector. So, we naturally include $w^{(i,j)}$ as a control variable and the relations in Proposition (2) as state constraints in the time optimal control. However, the inequality constraint in Proposition (2) has to be relaxed. No strict inequality can be written in the formulation of the optimal control problem. Moreover, the relational matrices, $S^{(i)}$, and the translational vectors $r^{(i)}$, $i = 1, \ldots, n_p$, are explicit functions of $x = (q, v)$ as explained in Appendix A. Consequently, the state constraints related to the collision avoidance are given for all $i = 1, \ldots, n_p$ and all $j = 1, \ldots, n_q$ by

$$\frac{A^{(i)} S^{(i)}(x(t))^T}{C^{(j)}} \hspace{1cm} w^{(i,j)}(t) = 0,$$

(4)

$$\frac{b^{(i)} + A^{(i)} S^{(i)}(x(t))^T r^{(i)}(x(t))}{d^{(j)}} \hspace{1cm} w^{(i,j)}(t) \leq -\varepsilon.$$

(5)
The constraints are called *anti-collision constraints* and $\varepsilon > 0$ is a given distance parameter. The complete formulation of the optimal control problem is given in the next section.

### 3. Optimal Control Problem

We combine the optimal control problem given in the introduction with the anti-collision constraints (4)-(5). This combination yields the model to find the optimal collision-free trajectory of a robot surrounded by an obstacle. First, we simplify the expression of the anti-collision constraints by using a new numbering of the constraints. We carry out the following transformation

$$I : \{1, \ldots, n_p\} \times \{1, \ldots, n_q\} \rightarrow \{1, \ldots, n_p n_q\}$$

$$(i, j) \mapsto I = (i - 1)n_q + j.$$  

To each pair $(i, j)$ there corresponds an index $I$ in $\{1, \ldots, n_p n_q\}$. Reciprocally, for each $I \in \{1, \ldots, n_p n_q\}$ only one pair $(i, j) \in \{1, \ldots, n_p\} \times \{1, \ldots, n_q\}$ exists.

In the sequel, the index $I$ is used instead of the pair $(i, j)$. The control variable $w^{(i,j)}$ is then numbered as $w_I$. Let us also define the functions $G_I : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{(p_i + q_j) \times 3}$ and $g_I : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{p_i + q_j}$ for $I = 1, \ldots, n_p n_q$ as follows

$$G_I(x) = \begin{pmatrix} A^{(i)} S^{(i)}(x)^T \\ C^{(j)} \end{pmatrix},$$
$$g_I(x) = \begin{pmatrix} b^{(i)} + A^{(i)} S^{(i)}(x)^T r^{(i)}(x) \\ d^{(j)} \end{pmatrix}.$$

Let us finally set:

- the number of indices $I$: $M = n_p n_q$;
- the size of $w_I$: $n_I = p_i + q_j$;
- the size of $w := (w_1, \ldots, w_M)$: $n_w = \sum_{I=1}^{M} n_I = n_Q \sum_{i=1}^{n_p} p_i + n_P \sum_{j=1}^{n_q} q_j$.

Then, after transformation onto the fixed time interval $T := [0, 1]$ and with these new notations the optimal control problem reads as follows:

(OCP): Minimize

$$\varphi(x(0), x(1), t_f)$$

with respect to $x \in W^{n_x}_{1,\infty}(T)$, $u \in L^{n_u}_{\infty}(T)$, $w \in L^{n_w}_{\infty}(T)$, and $t_f \geq 0$,

subject to the constraints

$$x'(t) - t_f f(x(t), u(t)) = 0 \quad a.e. \ in \ T,$$
$$\psi(x(0), x(1)) = 0,$$
$$G_I(x(t))^T w_I(t) = 0, \quad I = 1, \ldots, M, \ a.e. \ in \ T,$$
$$g_I(x(t))^T w_I(t) \leq -\varepsilon, \quad I = 1, \ldots, M, \ a.e. \ in \ T,$$
$$w_I(t) \geq 0, \quad I = 1, \ldots, M, \ a.e. \ in \ T,$$

and

$$u(t) \in \mathcal{U}.$$
As usual $L^\infty(T)$ denotes the Banach space of essentially bounded functions mapping from $T$ into $\mathbb{R}^{n_u}$ and $W^{1,\infty}_1(T)$ denotes the Banach space of absolutely continuous functions with essentially bounded derivative that map from $T$ into $\mathbb{R}^{n_x}$.

4. DIRECT DISCRETIZATION

We outline a direct approach towards the numerical solution of the collision avoidance problem (OCP) by means of discretization methods.

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. The approach is based on the grid $G_N := \{ t_k = t_0 + kh \mid k = 0, 1, \ldots, N \}$, which, for simplicity, is chosen equidistantly with the fixed step-size $h = (t_f - t_0)/N$.

4.1. Reduced Discretization. A reduced discretization is obtained by solving the discretized differential equations and eliminating them from the problem formulation. To this end we prefer to use explicit integration schemes.

A control parameterization is given by the B-spline representation

$$u_h(t; u_0, \ldots, u_{N+r-2}) := \sum_{i=0}^{N+r-2} u_i B_{ir}(t),$$

where $B_{ir}$, $i = 0, \ldots, N + r - 2$, denote elementary B-splines of order $r$ on $G_N$ and $(u_0, \ldots, u_{N+r-2})^T \in \mathbb{R}^{n_u(N+r-1)}$ is the vector of de Boor points. The order $r$ defines the smoothness of the control approximation, that is, $u_h$ is $r - 2$ times continuously differentiable for $r > 2$. Typically we use $r = 1$ ($u_N$ is piecewise constant) or $r = 2$ ($u_h$ is continuous and piecewise linear).

As the elementary B-splines of order $r$ sum up to one for all $t \in T$, the box constraints $u_h(t) \in \mathcal{U}$ are satisfied, if

$$u_i \in \mathcal{U}, \quad i = 0, \ldots, N + r - 2.$$

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 $B_{ir}$ have a local support only. The latter leads to certain sparsity patterns in the constraint Jacobian, which can be exploited numerically, see [11, 12].

Solving the differential equations for the initial value $x_0$ and a given $t_f$ by a one-step method with increment function $\Phi$, which typically defines an explicit Runge-Kutta method, leads to the state approximations

$$x_{k+1}(z) = x_k(z) + h\Phi(t_k, x_k(z), u_h(t_k; u_0, \ldots, u_{N+r-2}, t_f, h), k = 0, \ldots, N - 1,$$

at the grid points $t_k$, $k = 0, \ldots, N$. The state approximations depend on the vector

$$z := (x_0, u_0, \ldots, u_{N+r-2}, t_f)^T \in \mathbb{R}^{n_z},$$
with \( n_z = n_x + (N + r - 1)n_u + 1 \). In the simplest case we may choose the explicit Euler method with
\[
\Phi(t, x, u, t_f, h) := t_ff(x, u),
\]
but typically we prefer higher order methods like the classic Runge-Kutta method of order four. Similarly, the variables \( w_I, I = 1, \ldots, M \), are approximated by B-splines as follows
\[
w_{I,h}(t) = w_{I,h}(t; w_{I,0}, \ldots, w_{I,N+r-2}) := \sum_{i=0}^{N+r-2} w_{I,i} B_{ir}(t), \quad I = 1, \ldots, M.
\]
Introducing both, the control and state approximations, into the optimal control problem (OCP) leads to the following finite dimensional nonlinear optimization problem:

\[
\text{Minimize} \quad \varphi(x_0, x_N(z), t_f)
\]
with respect to \( z \in \mathbb{R}^{n_z} \) and \( w_{I,h}(t_k) \in \mathbb{R}^{n_I} \), for \( I = 1, \ldots, M \) and \( k = 0, \ldots, N \), subject to the constraints
\[
\begin{align*}
\psi(x_0, x_N(z)) &= 0, \\
w_{I,h}(t_k) &\geq 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
G_I(x_k(z))^T w_{I,h}(t_k) &= 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
g_I(x_k(z))^T w_{I,h}(t_k) &\leq -\varepsilon, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
w_k &\in U, \quad k = 0, \ldots, N.
\end{align*}
\]
Extensions towards a multiple shooting method by adding additional shooting nodes are possible as well. Note however that the reduced problem is still a sparse and potentially large-scale problem depending on the number \( M \) of anti-collision constraints.

### 4.2. Sequential Quadratic Programming (SQP)

From now on we choose to use B-splines of second order (i.e. \( r = 2 \)). Hence, we have
\[
w_{I,h}(t_k; w_{I,0}, \ldots, w_{I,N}) = w_{I,k}, \quad I = 1, \ldots, M, k = 0, \ldots, N.
\]
For simplicity, the notation \( w_{I,k} \) is preferred to \( w_{I,h}(t_k; w_{I,0}, \ldots, w_{I,N}) \) in the discretized formulation of the optimal control problem.

The reduced discretization method can be cast as an optimization problem with the following structure:

\[
\text{(NLP): Minimize } J(z) \text{ with respect to } z \in \mathbb{R}^{n_z} \text{ and } w = (w_{1,0}, \ldots, w_{1,N}, \ldots, w_{M,0}, \ldots, w_{M,N})^T \in \mathbb{R}^{(N+1)n_w}
\]
subject to the constraints

\[
\begin{align*}
    h(z) &= 0, \\
    w_{I,k} &\geq 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    \bar{G}_{I,k}(z)^T w_{I,k} &= 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    \bar{g}_{I,k}(z)^T w_{I,k} &\leq -\varepsilon, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    z &\in \mathcal{Z} := \{ z \in \mathbb{R}^n | z_\ell \leq z \leq z_u \}.
\end{align*}
\]

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.

**Remark 1.** The correspondence between the reduced discretization of (OCP) and the nonlinear optimization problem (NLP) can be seen as follows:

Let \( J(z) := \varphi(x_0, x_N(z), t_f) \) and \( h(z) := \psi(x_0, x_N(z)) \). For \( I = 1, \ldots, M, k = 0, \ldots, N \) define \( \bar{G}_{I,k} \) and \( \bar{g}_{I,k} \) by

\[
\bar{G}_{I,k}(z) := G_I(x_k(z)) \quad \text{and} \quad \bar{g}_{I,k}(z) := g_I(x_k(z)).
\]

The Lagrange function with multipliers \( \lambda, \mu = (\mu_1, \ldots, \mu_{M,N})^T, \eta = (\eta_1, \ldots, \eta_{M,N})^T, \sigma = (\sigma_1, \ldots, \sigma_{M,N})^T \) and \( \zeta \) read as

\[
L(z, w, \lambda, \mu, \eta, \sigma, \zeta) = J(z) + \lambda^T h(z) + \zeta^T z + \sum_{I=1}^M \sum_{k=0}^N \left[ \mu_{I,k}^T (-w_{I,k}) + \eta_{I,k} \bar{G}_{I,k}(z)^T w_{I,k} + \sigma_{I,k} \bar{g}_{I,k}(z)^T w_{I,k} \right].
\]

The Hessian of \( L \) with respect to \( (z, w) \) is given by

\[
L''_{(z,w),(z,w)} = \begin{pmatrix}
    L''_{zz} & \cdots & \left( \gamma'_{10,z} \right)^T \\
    \vdots & & \vdots \\
    \left( \gamma'_{MN,z} \right)^T & \cdots & \left( \gamma'_{MN,z} \right)^T \\
\end{pmatrix},
\]

where

\[
\gamma_{I,k}(z, \eta_{I,k}, \sigma_{I,k}) := \bar{G}_{I,k}(z) \eta_{I,k} + \bar{g}_{I,k}(z) \sigma_{I,k}, \quad I = 1, \ldots, M, k = 0, \ldots, N.
\]

A typical requirement in a convergence analysis or for checking second order sufficient conditions is that the Hessian matrix is positive definite on the null-space of the linearized active constraints. It is not at all clear that the above Hessian matrix satisfies this requirement owing to the zero block, which corresponds to \( L''_{ww} \). A remedy could be to regularize the problem by adding the term \( \alpha \| w \|^2 \) with a suitable value of \( \alpha \) to the objective function. Then the zero block is filled with \( \alpha I \). Such a regularization makes sense as the variable \( w \) apart from being feasible is arbitrary. In particular, if \( w \) is feasible, so is \( \beta w \) for every \( \beta \geq 1 \) owing to the linearity of the anti-collision constraints. Hence, norming \( w \) makes sense.

The following prototype SQP algorithm can be applied to the optimization problem. We restrict the discussion to the local SQP method only, but we point out that the algorithm needs to be

subject to the constraints

\[
\begin{align*}
    h(z) &= 0, \\
    w_{I,k} &\geq 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    \bar{G}_{I,k}(z)^T w_{I,k} &= 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    \bar{g}_{I,k}(z)^T w_{I,k} &\leq -\varepsilon, \quad I = 1, \ldots, M, k = 0, \ldots, N, \\
    z &\in \mathcal{Z} := \{ z \in \mathbb{R}^n | z_\ell \leq z \leq z_u \}.
\end{align*}
\]

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.

**Remark 1.** The correspondence between the reduced discretization of (OCP) and the nonlinear optimization problem (NLP) can be seen as follows:

Let \( J(z) := \varphi(x_0, x_N(z), t_f) \) and \( h(z) := \psi(x_0, x_N(z)) \). For \( I = 1, \ldots, M, k = 0, \ldots, N \) define \( \bar{G}_{I,k} \) and \( \bar{g}_{I,k} \) by

\[
\bar{G}_{I,k}(z) := G_I(x_k(z)) \quad \text{and} \quad \bar{g}_{I,k}(z) := g_I(x_k(z)).
\]

The Lagrange function with multipliers \( \lambda, \mu = (\mu_1, \ldots, \mu_{M,N})^T, \eta = (\eta_1, \ldots, \eta_{M,N})^T, \sigma = (\sigma_1, \ldots, \sigma_{M,N})^T \) and \( \zeta \) read as

\[
L(z, w, \lambda, \mu, \eta, \sigma, \zeta) = J(z) + \lambda^T h(z) + \zeta^T z + \sum_{I=1}^M \sum_{k=0}^N \left[ \mu_{I,k}^T (-w_{I,k}) + \eta_{I,k} \bar{G}_{I,k}(z)^T w_{I,k} + \sigma_{I,k} \bar{g}_{I,k}(z)^T w_{I,k} \right].
\]

The Hessian of \( L \) with respect to \( (z, w) \) is given by

\[
L''_{(z,w),(z,w)} = \begin{pmatrix}
    L''_{zz} & \cdots & \left( \gamma'_{10,z} \right)^T \\
    \vdots & & \vdots \\
    \left( \gamma'_{MN,z} \right)^T & \cdots & \left( \gamma'_{MN,z} \right)^T \\
\end{pmatrix},
\]

where

\[
\gamma_{I,k}(z, \eta_{I,k}, \sigma_{I,k}) := \bar{G}_{I,k}(z) \eta_{I,k} + \bar{g}_{I,k}(z) \sigma_{I,k}, \quad I = 1, \ldots, M, k = 0, \ldots, N.
\]

A typical requirement in a convergence analysis or for checking second order sufficient conditions is that the Hessian matrix is positive definite on the null-space of the linearized active constraints. It is not at all clear that the above Hessian matrix satisfies this requirement owing to the zero block, which corresponds to \( L''_{ww} \). A remedy could be to regularize the problem by adding the term \( \alpha \| w \|^2 \) with a suitable value of \( \alpha \) to the objective function. Then the zero block is filled with \( \alpha I \). Such a regularization makes sense as the variable \( w \) apart from being feasible is arbitrary. In particular, if \( w \) is feasible, so is \( \beta w \) for every \( \beta \geq 1 \) owing to the linearity of the anti-collision constraints. Hence, norming \( w \) makes sense.

The following prototype SQP algorithm can be applied to the optimization problem. We restrict the discussion to the local SQP method only, but we point out that the algorithm needs to be
augmented by a globalization strategy in order to achieve numerical robustness and convergence from remote starting points. As in [27] we used an Armijo type line-search procedure for the augmented Lagrangian function in our implementation.

Prototype local SQP method:

(0) Choose \( z^{(0)} \in Z, w^{(0)} \geq 0, \nu^{(0)} := (\lambda^{(0)}, \mu^{(0)}, \eta^{(0)}, \sigma^{(0)}, \zeta^{(0)}), B_0 := L''_{(z,w),(z,w)}(z^{(0)}, w^{(0)}, \nu^{(0)}), \) and set \( \ell := 0. \)

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

(2) Compute a KKT point \((d^{(\ell)}, \nu^{(\ell+1)})\) of the following linear-quadratic optimization problem:

Minimize \[
\frac{1}{2} d^T B_\ell d + J'(z^{(\ell)}) dz
\]

with respect to \(d = (d_z, d_{w_{1,0}}, \ldots, d_{w_{M,N}})^T\) subject to the constraints

\[
h(z^{(\ell)}) + h'(z^{(\ell)}) dz = 0, \quad w^{(\ell)} + d_{w_{1,k}} \geq 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \]

\[
\bar{G}_{I,k}^I(z^{(\ell)})^T w^{(\ell)} + \bar{G}_{I,k}^I(z^{(\ell)})^T d_{w_{1,k}} + \bar{G}_{I,k}^I(z^{(\ell)})^T (w^{(\ell)} + d_{z}) = 0, \quad I = 1, \ldots, M, k = 0, \ldots, N, \]

\[
\bar{g}_{I,k}^I(z^{(\ell)})^T w^{(\ell)} + \bar{g}_{I,k}^I(z^{(\ell)})^T d_{z} \leq -\epsilon, \quad I = 1, \ldots, M, k = 0, \ldots, N, \]

\[z^{(\ell)} + d_z \in Z.\]

(3) Set \[z^{(\ell+1)} := z^{(\ell)} + d_z, \quad w_{I,k}^{(\ell+1)} := w_{I,k}^{(\ell)} + d_{w_{1,k}}^{(\ell)}, \quad I = 1, \ldots, M, k = 0, \ldots, N,\]

and compute \[B_{\ell+1} := L''_{(z,w),(z,w)}(z^{(\ell+1)}, w^{(\ell+1)}, \nu^{(\ell+1)}).\]

Set \(\ell := \ell + 1,\) and go to (1).

Remark 2.

(a) If the constraints of the quadratic subproblem turn out to be infeasible, the quadratic subproblem is replaced by one with relaxed constraints, see [4, 24]. A convergence analysis for an SQP method using the augmented Lagrange function can be found in [26, 27].

(b) In order to achieve convergence from remote starting points, the algorithm has to be augmented by a globalization strategy, see [24, 27] for line-search based methods and [8, 9] for filter methods.

(c) Instead of the exact Hessian matrix one can use BFGS update formulas for \(B_\ell,\) which guarantee that \(B_\ell\) is symmetric and positive definite and thus the quadratic subproblems are strictly convex. Special measures for handling sparse matrices and sparse Hessian updates have to be taken for large problems, see [2].

(d) There are powerful algorithms for the numerical solution of quadratic optimization problems using primal or dual active set methods, see [17, 18, 19, 28], or interior-point methods, see [14, 29]. The method in [14] is designed for large-scale and sparse problems.
5. **Backface Culling Active Set Strategy**

The discretized optimal control problem described in the previous section contains a large number of state constraints. At each time step \( t_k, k = 0, \ldots, N \), and for every pair of polyhedra \((P^{(i)}, Q^{(j)})\), four collision avoidance constraints are defined. This yields \( 4M(N+1) \) state constraints in (NLP). This large number could be problematic to find the solution of (NLP) as outlined in [23]. Are all these constraints necessary? Let us assume that at the time step \( t_k \) the obstacle is located far from the robot. In that case, the robot can move without colliding with the obstacle. The collision avoidance constraints defined at \( t_k \) are then unnecessary. In the same way, if the obstacle is located behind the robot at \( t_k \), the robot can go forward without collision. The related state constraints are again superfluous. Furthermore, only few of the faces of the convex polyhedra do actually collide while most faces are remotely located on the opposite side of the object. This selection of visible obstacles and faces is called **backface culling**, see [10, 25, 30].

Backface culling is mostly used in computer graphics. Vaněček Jr. in [30] first suggested to use this technique for collision detection. However, his criterion to select the visible faces works only between polyhedra which face each other. Here, we develop other criteria. They all concern the position of the robot relative to the obstacles. The first criterion looks at the distance between the robot and the obstacles. If this distance is large, then no collision will occur and there is no need to consider the collision avoidance criterion for this case.

5.1. **Criteria.** In this section \((P, Q)\) denotes a pair of polyhedra. The polyhedron \( P \) belongs to the approximation of the robot and \( Q \) refers to the obstacle. In comparison to the previous sections, we drop the exponent \((i, j)\) for more readability. Moreover, the figures in this section are given for a two dimensional case, even if the criteria are established for two or three dimensional domain. Finally, the criteria must be fast and easy to compute because they will be employed at each intermediate SQP iteration, between each pair of polyhedra (i.e. for \( I = 1, \ldots, M \)) and at each time step \( t_k, K = 0, \ldots, N \).

The first criterion check if \( Q \) is far from \( P \). The distance between \( P \) and \( Q \) is roughly computed. A rectangle surrounded box is defined for each polyhedron. The intersection between the boxes is then easily tested. If the intersection is empty, it means that \( Q \) is distant from \( P \) and the anti-collision constraint is not taken in consideration for the pair \((P, Q)\).

In order to compute these boxes, let us denote

- \((x^P_i, y^P_i, z^P_i), i = 1, \ldots, s^P\), the vertices of \( P \) with \( s^P \) the number of vertices,
- \((x^Q_i, y^Q_i, z^Q_i), i = 1, \ldots, s^Q\), the vertices of \( Q \) with \( s^Q \) the number of vertices,

and define for \( P \) and \( Q \)

\[
\begin{align*}
x^m_P &= \min_{i=1, \ldots, s^P} x^P_i, & y^m_P &= \min_{i=1, \ldots, s^P} y^P_i, & z^m_P &= \min_{i=1, \ldots, s^P} z^P_i, \\
x^M_P &= \max_{i=1, \ldots, s^P} x^P_i, & y^M_P &= \max_{i=1, \ldots, s^P} y^P_i, & z^M_P &= \max_{i=1, \ldots, s^P} z^P_i, \\
x^m_Q &= \min_{i=1, \ldots, s^Q} x^Q_i, & y^m_Q &= \min_{i=1, \ldots, s^Q} y^Q_i, & z^m_Q &= \min_{i=1, \ldots, s^Q} z^Q_i, \\
x^M_Q &= \max_{i=1, \ldots, s^Q} x^Q_i, & y^M_Q &= \max_{i=1, \ldots, s^Q} y^Q_i, & z^M_Q &= \max_{i=1, \ldots, s^Q} z^Q_i.
\end{align*}
\]
The tuple \((x_m, y_m, z_m, x_M, y_M, z_M)\), resp. \((x_Q, y_Q, z_Q, x_Q, y_Q, z_Q)\), defines the smallest rectangle box around \(P\), resp. \(Q\). For the criterion a larger box is considered. Let \(\delta > 0\). Let \(B^P\), resp. \(B^Q\), denotes the box generated by the tuple \((x_m - \delta, y_m - \delta, z_m - \delta, x_M + \delta, y_M + \delta, z_M + \delta)\), resp. \((x_Q - \delta, y_Q - \delta, z_Q - \delta, x_Q + \delta, y_Q + \delta, z_Q + \delta)\). In other words, \(B^P\) and \(B^Q\) are given by

\[
B^P = [x_m - \delta, x_M + \delta] \times [y_m - \delta, y_M + \delta] \times [z_m - \delta, z_M + \delta]
\]

\[\text{(6)}\]

\[
B^Q = [x_Q - \delta, x_Q + \delta] \times [y_Q - \delta, y_Q + \delta] \times [z_Q - \delta, z_Q + \delta].
\]

An illustration in a 2 dimensional workspace is given in Figure 2. The boxes \(B^P\) and \(B^Q\) are separated if

\[
[x_m - \delta, x_M + \delta] \cap [x_Q - \delta, x_Q + \delta] = \emptyset
\]

or

\[
[y_m - \delta, y_M + \delta] \cap [y_Q - \delta, y_Q + \delta] = \emptyset
\]

or

\[
[z_m - \delta, z_M + \delta] \cap [z_Q - \delta, z_Q + \delta] = \emptyset
\]

This relation is equivalent to

\[
x_Q - \delta > x_M + \delta \quad \text{or} \quad x_Q + \delta < x_M - \delta
\]

\[\text{(7)}\]

\[
y_Q - \delta > y_M + \delta \quad \text{or} \quad y_Q + \delta < y_M - \delta
\]

or

\[
z_Q - \delta > z_M + \delta \quad \text{or} \quad z_Q + \delta < z_M - \delta
\]

If at least one of the above strict inequalities is satisfied, then \(B^P\) does not intersect \(B^Q\). It follows that \(Q\) is remote and the anti-collision constraint is not considered for the pair \((P, Q)\). So, the first criterion reads

**Criterion 1**

Build the boxes \(B^P\) and \(B^Q\) of \(P\) and \(Q\) according to (6). If one of the inequalities in (7) is satisfied, then no anti-collision constraint is written for the pair \((P, Q)\).

For the example in Figure 2, the second inequality of (7) is satisfied. The boxes \(B^P\) and \(B^Q\) are separated and according to Criterion 1 no anti-collision constraint is considered.

**Remark 3.** The coordinates of the vertices of \(P\) evolve in time since they belong to the robot. Hence the box \(B^P\) has to be determined at each grid point \(t_k\), \(k = 0, \ldots, N\). Because the obstacle does not move, the box \(B^Q\) is computed only once.

Let us assume now that the polyhedron \(Q\) is close enough to \(P\). The next criteria will determine which faces of \(Q\) are visible by \(P\). Consider the situation depicted in Figure 3 (a): the polyhedron \(P\) is moving downwards and \(v_c\) indicates the velocity of the center of gravity.

In the definition \(Q = \{x \in \mathbb{R}^2 \mid Cx \leq d\}\), each line of the system of inequalities describes a halfspace. The union of all the halfspaces generates the polyhedron \(Q\) and each face of the polyhedron is a piece of the associated hyperplane. Let us recall the collision avoidance criterion between the two polyhedra \(P\) and \(Q\): \(P\) and \(Q\) do not collide if and only if

\[
\exists w > 0, \text{ such that } \begin{pmatrix} A \\ C \end{pmatrix}^T w = 0 \text{ and } \begin{pmatrix} b \\ d \end{pmatrix}^T w < 0.
\]

\[\text{(8)}\]

We can observe that each component of \(w\) is multiplying one line of \(A\) or \(C\), and a component of \(b\) or \(d\). Hence each component of \(w\) is associated to a face of \(P\) or \(Q\).
Let us define the set \( \tilde{Q} \) generated by the faces \( e_1 \) and \( e_2 \) of \( Q \). \( \tilde{Q} \) is represented in Figure 3 (b). In fact \( e_1 \) and \( e_2 \) are the faces of \( Q \) visible by \( P \). According to (8) \( \tilde{Q} \) do not collide with \( P \) if and only if

\[
\exists \tilde{w} > 0, \text{ such that } \begin{pmatrix} A \\ C_{1,2} \end{pmatrix}^T \tilde{w} = 0 \text{ and } \begin{pmatrix} b \\ d_{1,2} \end{pmatrix}^T \tilde{w} < 0,
\]

where \( C_{1,2} \) is the matrix composed of the first two lines of \( C \) and \( d_{1,2} \) is the vector composed of the first two components of \( d \).

If \( \tilde{w} \) exists, then by setting \( w = (\tilde{w}, 0, 0, 0) \) the criterion (8) is satisfied. It follows that \( P \) and \( Q \) do not collide. In conclusion if no collision occurs between \( \tilde{Q} \) and \( P \), then \( Q \) and \( P \) do not collide. In (9) only the faces visible by \( P \) are taken in consideration. The dimension of \( \tilde{w} \) is always smaller than \( w \), because the polyhedra are supposed to be compact. Hence the problem to find \( \tilde{w} \) is always smaller.
In the sequel we will consider the collision avoidance criterion (9) defined only with the visible faces of $Q$, (9) being faster to verify than (8). The next criteria concern the determination of the visible faces of $Q$ relatively to $P$.

The faces of $Q$ which are located behind $P$ are invisible for $P$. To find them we have first to determine which vertex of $P$ is located the most in the opposite direction of the velocity of the center of gravity. This vertex is denoted by $S_R$. Then we consider the lower halfspace generated by the normal vector $v_c$ and the point $S_R$. All faces of $Q$ located in this halfspace are invisible.

The vertex $S_R$ is the vertex of $P$ which minimizes the scalar product between the vertices of $P$ and $v_c$. In other words, we have

$$S_R = \arg \min_{i=1,\ldots,p} v_i^T OS_i,$$

where $O$ is the origin and $S_i$, $i = 1, \ldots, p$, are the vertices of $P$. The equation of the lower halfspace is then given by

$$\mathcal{H}_{S_R} = \{ x \in \mathbb{R}^3 \mid v_c^T (x - S_R) < 0 \}.$$

A face $e$ of $Q$ is located behind $P$ if $e$ is included in the lower-halfspace $\mathcal{H}_{S_R}$. Hence, if $S_{e,i}$, $i = 1, \ldots, s_e^e$, denote the vertices of $e$, we have to test if

$$v_c^T (S_{e,i} - S_R) < 0, \forall i = 1, \ldots, s_e^e$$

or equivalently, to check if

$$\max_{i=1,\ldots,s_e^e} (v_c^T (S_{e,i} - S_R)) < 0.$$

![Figure 4](image-url)

**Figure 4.** (a) The faces $e_3$ and $e_4$ of $Q$ are located behind $P$. (b) All faces of $Q$ are located behind $P$.

Then the second criterion reads

**Criterion 2** A face $e$ of $Q$ composed by the vertices $S_{e,1}, \ldots, S_{e,s_e^e}$ is invisible by $P$ if

$$\max_{i=1,\ldots,s_e^e} v_c^T (S_{e,i} - S_R) < 0.$$

An example is given in Figure 4-(a) where the faces $e_3$ and $e_4$ satisfy Criterion 2. If all faces of $Q$ are located behind $P$ as illustrated in Figure 4-(b), $Q$ is not visible by $P$ and the anti-collision
constraint is not considered. This leads to

**Criterion 3** If all faces of \( Q \) are invisible by \( P \) according to Criterion 2, then no anti-collision constraint is written for the pair of polyhedra \((P, Q)\).

**Remark 4.** We supposed for the computation of \( S_R \) that the velocity of the center of gravity was not equal to zero. If the velocity is null, Criteria 2 and 3 are not applied.

Not all remaining faces of \( Q \) are visible by \( P \). Some of them can be hidden by other faces of \( Q \). It is the case for the face \( e_5 \) in Figure 3-(b). Let consider the Figure 5-(a). In this figure the vertex \( S \) of \( P \) can see the face \( e_5 \) of \( Q \). The hatched triangle \( BSA \) corresponds to the view angle of \( S \). In Figure 5-(b) the triangle \( BSA \) crosses the face \( e_1 \) of \( Q \). The vertex \( S \) cannot see \( e_5 \) because the face is hidden by \( e_1 \). The face \( e_5 \) is no more visible by \( S \) as soon as \( S \) is located in the halfspace \( \mathcal{H}_5^{-} = \{ x \in \mathbb{R}^3 \mid C_5 x < d_5 \} \) generated by \( e_5 \). From this observation comes the last criterion which determines which faces of \( Q \) are hidden for \( P \). The fourth criterion reads as follows

**Criterion 4** The face \( e_i \) of \( Q \) is invisible by the polyhedron \( P \) if

\[
P \subset \mathcal{H}_i^{-} = \{ x \in \mathbb{R}^3 \mid C_i x < d_i \},
\]

which is equivalent to require that

\[
\max_{j=1,\ldots,s^P}(C_iS_j - d_i) < 0,
\]

where \( S_j, j = 1, \ldots, s^P \) denote the vertices of \( P \).

---

**Figure 5.** (a) The vertex \( S \), and consequently the polyhedron \( P \), can see the face \( e_5 \). (b) The face \( e_5 \) of \( Q \) is invisible by \( P \) because \( P \subset \mathcal{H}_5^{-} \).

A limit case exists with Criterion 4 when \( P \) is included in \( Q \) as depicted in Figure 6. In that case all faces of \( Q \) are invisible for \( P \) according to Criterion 4. But in fact all these faces must...
be considered in the anti-collision constraint. So, in this particular case Criterion 4 must not be applied.

**Remark 5.** Criterion 4 can also be applied to detect which faces of \( P \) are visible by \( Q \). Then the anti-collision constraint defined for the pair \((P, Q)\) can be reduced as it was done in (9).

In this section criteria to determine the visible faces of \( Q \) were developed, provided \( Q \) is visible. The four criteria depend on the position of \( P \) (given by \( q \)) and also on the velocity of \( P \) for Criteria 2 and 3 (given by \( v \)). In the next section we show how the backface culling strategy is included in the SQP algorithm to solve (NLP).

### 5.2. Algorithm

Let us recall the index transformation that associates at each pair \((i, j)\) the new index \( I \) via the formula: \( I = (i - 1)n_q + j \). Let us also define the two following sets of pair of indices

\[
S := \{ (I, k) \mid I = 1, \ldots, M \text{ and } k = 0, \ldots, N \},
\]

\[
K := \{ (I, k) \in S \mid \text{the polyhedron } Q^{(j)} \text{ is visible by } P^{(i)} \text{ at } t_k \}.
\]

Hence, \( K \) is a subset of \( S \) and it is determined by applying the Criteria 1 and 3 established in the previous subsection.

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 \( Q^{(j)} \). In other words, the \( c^{th} \) component of \( w_I \) corresponds to

- the face \( c \) of \( P^{(i)} \) if \( 1 \leq c \leq p_i \),
- or the face \( c-p_i \) of \( Q^{(j)} \) if \( p_i+1 \leq c \leq p_i+q_j \).

Hence, let us define the following set of indices for each \((I, k) \in K\)

\[
J_{I,k} := \{ c \in \{1, \ldots, p_i\} \mid \text{the face } c \text{ of } P^{(i)} \text{ is invisible by } Q^{(j)} \text{ at } t_k \}
\]

\[
\cup \{ c \in \{p_i+1, \ldots, p_i+q_j\} \mid \text{the face } c \text{ of } Q^{(j)} \text{ is invisible by } P^{(i)} \text{ at } t_k \}.
\]

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 with the Criteria 2 and 4 of the previous subsection.

Backface culling consists of considering the anti-collision constraints whose pair of indices \((I, k) \in K\) and write these constraints only with the components of the variables
related to the visible faces. The algorithm to solve the \((NLP)\) problem 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**

1. Choose \(\varepsilon > 0\), \(z^{(0)} \in Z\), \(w^{(0)} \geq 0\) and \(\nu^{(0)} := (\lambda^{(0)}, \mu^{(0)}, \eta^{(0)}, \sigma^{(0)}, \zeta^{(0)})\).
2. Determine the sets of indices \(\mathcal{K}^{(0)}\) and \(\mathcal{J}^{(0)}_{\ell,k}\) for all \((I, k) \in \mathcal{K}^{(0)}\).
3. Compute \(B_0 := L^\nu_{(z,w),(z,w)}(z^{(0)}, w^{(0)}, \nu^{(0)})\) and set \(\ell := 0\).
4. If \((z^{(\ell)}, w^{(\ell)}, \nu^{(\ell)})\) is a KKT point of the optimization problem, STOP.
5. Compute a KKT point \((d^{(\ell)}, \nu^{(\ell+1)}_{\mathcal{K}^{(\ell)}})\) with
   \[
   \nu^{(\ell+1)}_{\mathcal{K}^{(\ell)}} := (\lambda^{(\ell+1)}, \mu^{(\ell+1)}_{\mathcal{K}^{(\ell)}}, \eta^{(\ell+1)}_{\mathcal{K}^{(\ell)}}, \sigma^{(\ell+1)}_{\mathcal{K}^{(\ell)}}, \zeta^{(\ell+1)})
   \]
   and
   \[
   \mu^{(\ell+1)}_{\mathcal{K}^{(\ell)}} := (\mu^{(\ell+1)}_{I,k})_{(I,k) \in \mathcal{K}^{(\ell)}},
   \eta^{(\ell+1)}_{\mathcal{K}^{(\ell)}} := (\eta^{(\ell+1)}_{I,k})_{(I,k) \in \mathcal{K}^{(\ell)}},
   \sigma^{(\ell+1)}_{\mathcal{K}^{(\ell)}} := (\sigma^{(\ell+1)}_{I,k})_{(I,k) \in \mathcal{K}^{(\ell)}},
   \]
   of the following linear-quadratic optimization problem:

   **Minimize**
   \[
   \frac{1}{2} d^T B_{\ell} d + J'(z^{(\ell)}) d_z
   \]
   with respect to \(d_z\) and \(d_{w_{I,k}}\), \((I, k) \in \mathcal{K}^{(\ell)}\), subject to the constraints
   \[
   h(z^{(\ell)}) + h'(z^{(\ell)}) d_z = 0,
   w_{I,k}^{(\ell)} + d_{w_{I,k}} \geq 0, \quad (I, k) \in \mathcal{K}^{(\ell)},
   \bar{G}_{I,k} (z^{(\ell)})^T w_{I,k}^{(\ell)} + \bar{G}_{I,k} (z^{(\ell)})^T d_{w_{I,k}}
   + \bar{G}'_{I,k} (z^{(\ell)})^T (w_{I,k}^{(\ell)}, d_z) = 0, \quad (I, k) \in \mathcal{K}^{(\ell)},
   \bar{g}_{I,k} (z^{(\ell)})^T w_{I,k}^{(\ell)} + \bar{g}_{I,k} (z^{(\ell)})^T d_{w_{I,k}}
   + \bar{g}'_{I,k} (z^{(\ell)})^T (w_{I,k}^{(\ell)}, d_z) \leq -\varepsilon, \quad (I, k) \in \mathcal{K}^{(\ell)},
   z^{(\ell)} + d_z \in Z,
   d_{w_{I,k}} = 0, \quad c \in \mathcal{J}^{(\ell)}_{I,k}, (I, k) \in \mathcal{K}^{(\ell)}.
   \]

   Note: The constraints \(d_{w_{I,k}} = 0\) are only included for notational simplicity. In practice these variables are actually eliminated from the problem formulation and thus no Lagrange multipliers for these constraints are foreseen in the multiplier vector \(\nu^{(\ell+1)}_{\mathcal{K}^{(\ell)}}\).

6. Set
   \[
   z^{(\ell+1)} := z^{(\ell)} + d_z, \quad w_{I,k}^{(\ell+1)} := w_{I,k}^{(\ell)} + d_{w_{I,k}}^{(\ell)}, \quad (I, k) \in \mathcal{K}^{(\ell)}.
   \]
Let $\nu^{(\ell+1)}$ be the vector with $\nu^{(\ell+1)}_{K^{(\ell)}}$ sorted in at the appropriate positions and with the remaining entries filled up with zeros.

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

6. NUMERICAL RESULT

Let us consider the robot presented in Figure 11 given in Appendix A. The robotic arm is composed by a socket, two links and three revolute joints. The socket rotates along the $z$-axis. The links rotate along the $y$-axis. At the end of the second link a load is fixed. We consider one obstacle which is represented in Figure 7. The robot is asked to move the load around the obstacle. For this numerical example, the collision avoidance is only applied between the load and the obstacle.

![Figure 7. A 3-link robot with a load next to an obstacle.](image)

The load is a cube. Its representation as a system of linear inequalities is given by

$$P = \{ x \in \mathbb{R}^3 \mid Ax \leq b \},$$

with

$$A = \begin{pmatrix}
1 & 0 & 0 \\
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1 \\
0 & 0 & -1
\end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix}
0.25 \\
0.25 \\
0.25 \\
0.25 \\
0.25 \\
0.25
\end{pmatrix},$$

where the origin of the system of axis is located at the center of the cube. Similarly, the obstacle is given by:

$$Q = \{ x \in \mathbb{R}^3 \mid Cx \leq d \},$$

where $C = A$ and $d^T = (2, 2, 1.4, -1, 4.5, -1.5)$.

The definition of the obstacle is given in the reference frame.

The following parameters of the robot are used for the numerical computations.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m₁</td>
<td>10</td>
<td>mass of socket</td>
</tr>
<tr>
<td>m₂</td>
<td>2</td>
<td>mass of link 1</td>
</tr>
<tr>
<td>m₃</td>
<td>2</td>
<td>mass of link 2</td>
</tr>
<tr>
<td>m₄</td>
<td>1</td>
<td>mass of load</td>
</tr>
<tr>
<td>h₁</td>
<td>1</td>
<td>height of socket</td>
</tr>
<tr>
<td>l₁</td>
<td>1</td>
<td>length of link 1</td>
</tr>
<tr>
<td>l₂</td>
<td>1</td>
<td>length of link 2</td>
</tr>
<tr>
<td>r₁</td>
<td>0.1</td>
<td>radius of socket and links</td>
</tr>
<tr>
<td>r₂</td>
<td>0.3</td>
<td>radius of second socket</td>
</tr>
<tr>
<td>r₃</td>
<td>0.5</td>
<td>radius of platform</td>
</tr>
<tr>
<td>r₄</td>
<td>0.1</td>
<td>radius of load</td>
</tr>
<tr>
<td>b₁</td>
<td>0</td>
<td>offset of link 1</td>
</tr>
<tr>
<td>b₂</td>
<td>0</td>
<td>offset of link 2</td>
</tr>
</tbody>
</table>

The moment of inertia of the socket, the links and the load are respectively given by

$$J₁ = \text{diag}(0, 0, \frac{1}{2}m₁r₁²),$$
$$J₂ = \text{diag}(\frac{1}{4}m₂r₁², \frac{1}{2}m₃r₁², \frac{1}{4}m₄r₁²),$$
$$J₃ = \text{diag}(\frac{1}{4}m₃r₃², \frac{1}{2}m₄r₃², \frac{1}{4}m₅r₃²),$$
$$J₄ = \text{diag}(\frac{2}{5}m₄r₄², \frac{2}{5}m₅r₄², \frac{2}{5}m₆r₄²).$$

To solve the optimal control problem (NLP) with the SQP method, we choose 21 control grid points and $\varepsilon = 10^{-5}$ for the anti-collision constraints. The ordinary differential equations are integrating with the classical Runge-Kutta method (RK4) and the control variables $u$ and $w$ are approximated with B-splines of second order (i.e. $r = 2$).

In this example, the obstacle is always located close enough to the robot and never behind it. Consequently, the number of state constraints is not reduced with the backface culling strategy. However, the anti-collision constraints are written in a reduced form, i.e. they only contain the components of $w$ related to the visible faces of the robot and of the obstacle. So, with this example, we can appreciate the reduction of the number of unknowns and the simplification of the anti-collision constraints.

Some snapshots of the robot motion are given in Figure 8. A face of the obstacle is black, if the face satisfies Criteria 2 and 4. Otherwise, the face is grey. From $t_1$ to $t_5$ the left face and the bottom face are the sole visible faces. Between $t_6$ and $t_9$ the right face is also visible. From $t_9$, the left face is culled and from $t_15$ only the right face remains visible. So, at most 3 faces are not deleted. The faces on the side (except the bottom face) are always invisible. Hence, the corresponding components in the control $w$ can be eliminated. This means that the number of unknowns is 50% less with the backface culling strategy.

The time evolution of the control variables $u$ and $w$ are represented in Figures 9 and 10. We can observe a bang-bang-like behavior for $u$. This behavior is expected and was already observed by [31] for robot motion planning. The control $w$ do not follow a bang-bang phenomenon. In Figure 10 (a) the first six components of $w$ are plotted. These components correspond to the
faces of the load. In Figure 10 (b) only the components related to the visible faces of the obstacle are plotted. We have the relation:

- \( w_9 \): the left face,
- \( w_{10} \): the right face,
- \( w_{12} \): the bottom face.

We can see that the component \( w_{10} \) becomes positive at \( t_6 \). This corresponds to the new visibility of the right face from \( t_6 \). We can also remark that this event on \( w_{10} \) has an impact on the other components of \( w \). From \( t_1 \) to \( t_5 \), \( w_4 \) and \( w_{12} \) were increasing. The event at \( t_6 \) changed their behavior. Both started then to decrease. A similar phenomenon can be observed on the variable \( w_1, w_2, w_5 \) and \( w_9 \); the trajectory of these variables is perturbed at \( t_6 \).

At \( t_9 \), the component \( w_9 \) becomes null and the left face is no more visible by the robot. Note that between \( t_6 \) and \( t_9 \) all the positive components of \( w \) were decreasing and tending to 0. Once \( w_9 \) becomes null, the other components of \( w \) start to increase again. The last event occurs at \( t_{15} \), when \( w_{12} \) is equal to 0. An influence of the activation on the evolution of the other variables is observable.
Finally, let us consider Table 1. In this table, we compare the results obtained with and without the backface culling strategy. The final time, the number of iterations of the SQP method and the CPU times are stored in this table. In the first column, the values are obtained by solving the discretized optimal control problem without backface culling. The second column is obtained with the algorithm depicted in Section 5.

First, let us note that the final times \( t_f \) are equal. The trajectories are also similar. So, we can validate the solution obtained with the backface culling strategy. The number of iterations is smaller for the backface culling strategy: we need 530 iterations instead of 813. The gain of the backface culling is even better when we compare the CPU times: almost 4 minutes for a traditional resolution and less than 1 minute for the backface culling. So, we are almost 4 times faster. This is due on one side to the smaller number of iterations and on the other side to the smaller size of the problem to solve: less unknowns and reduced constraints.
<table>
<thead>
<tr>
<th></th>
<th>Global</th>
<th>Backface culling</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_f$</td>
<td>0.4261s</td>
<td>0.4261s</td>
</tr>
<tr>
<td>Iter.</td>
<td>813</td>
<td>530</td>
</tr>
<tr>
<td>CPU</td>
<td>3 min 44.33s</td>
<td>51.61 s</td>
</tr>
</tbody>
</table>

**Table 1.** Comparison of $t_f$, the number of iterations in the SQP method and the CPU times with and without the backface culling strategy.

The numerical solution of the collision avoidance problems turned out to be challenging. The SQP method typically has difficulties to satisfy the constraints, if insufficient initial guesses are provided. Appropriate scaling was necessary as well. A remedy is to shift some of the constraints into the objective function using a penalty term. In our experiments we have chosen the terminal boundary conditions and added the penalty term

$$\alpha \| x(t_f) - x_f \|^2$$

with $\alpha > 0$ sufficiently large to the objective function. The code then appears to be more robust, but requires more iterates.

**7. Conclusion and Outlook**

A time optimal control problem to find for a robot the fastest trajectory that avoids collision with surrounding obstacles was established. The robot and the obstacles were represented as finite unions of convex compact polyhedra. This description combined with Farkas's lemma allowed us to express the collision avoidance as state constraints. A backface culling strategy was built to apply the anti-collision constraints only between the visible part of polyhedra which can see each other. This strategy was finally added in the SQP as an active set strategy. Numerical results illustrated the gain of such a strategy in the resolution of (NLP).

Eventually, we outline an approach that can be used to further reduce or reformulate the collision avoidance optimal control problems.

**7.1. Elimination of Equality Constraints.** The rank of the matrices $G_I(x) \in \mathbb{R}^{(p_i+q_j) \times 3}$ is supposed to be 3, which can be guaranteed for typical geometric objects like boxes or tetrahedra. Hence, the equality constraints

$$G_I(x(t))^\top w_I(t) = 0, \quad I = 1, \ldots, M,$$

can be considered index-one algebraic constraints, where 3 components of $w_I$ are considered algebraic variables. To this end, let $J \subset \{1, \ldots, p_i + q_j\}$ and $J := \{1, \ldots, p_i + q_j\} \setminus J$ be index sets such that $G^I_J(x)$ is non-singular, where $G^I_J(x)$ denotes those rows of $G_I(x)$ whose index belongs to the index set $J$. For simplicity we assume that $J$ does not depend on the index $I$, that is, all geometric objects are described by the same matrices. By solving the algebraic constraints we obtain

$$G^I_J(x)^\top w^I_J + G^I_J(x)^\top w^I_J = 0 \iff w^I_J = -G^I_J(x)^\top G^I_J(x)^\top w^I_J.$$

Define $\bar{w}_I := w^I_J$ and reduced matrices

$$\bar{G}_I(x)^\top := G^I_J(x)^\top G^I_J(x)^\top,$$

$$\bar{g}_I(x)^\top := g^I_J(x)^\top - g^I_J(x)^\top \bar{G}_I(x)^\top.$$
Taking into account the non-negativity of $w_I$ for $I = 1, \ldots, M$, we obtain the following reduced optimal control problem with mixed control-state inequality constraints:

\[
\text{Minimize} \quad \varphi(x(0), x(1), t_f)
\]

with respect to $x \in W_{1,\infty}^n(T)$, $u \in L_{\infty}^n(T)$, $\bar{w} \in L_{\infty}^n(T)$, and $t_f \geq 0$ subject to the constraints

\[
x'(t) - t_f f(x(t), u(t)) = 0 \quad \text{a.e. in } T,
\]

\[
\psi(x(0), x(1)) = 0,
\]

\[
\tilde{g}_I(x(t))^T \bar{w}_I(t) \leq -\varepsilon, \quad I = 1, \ldots, M, \text{ a.e. in } T,
\]

\[
\tilde{G}_I(x(t))^T \bar{w}_I(t) \leq 0, \quad I = 1, \ldots, M, \text{ a.e. in } T,
\]

and

\[
u(t) \in \mathcal{U} := \{u \in \mathbb{R}^{n_u} \mid u_{\text{min}} \leq u \leq u_{\text{max}}\}.
\]

This problem is smaller than the original problem and instead of the equality constraints resulting from the anti-collision constraints it only has inequality constraints. It turned out in numerical experiments that the latter problem seems to be more robust with regard to the choice of the initial guess. Moreover, Criteria 1 and 3 of the backface culling can still be applied to select the pairs of polyhedra for which the anti-collision constraints are defined. This is not the case for Criteria 2 and 4. If only the visible faces are considered in the anti-collision constraints, then the index set $J$ may not exists.

### 7.2. Elimination of Artificial Control Variables

An alternate approach to handle anti-collision constraints without artificial control variables uses the following parametric linear programs for $I = 1, \ldots, M$:

\[
\text{LP}_I(x) \quad \text{Minimize} \quad g_I(x)^T w \quad \text{w.r.t. } w \text{ subject to } G_I(x)^T w = 0, \ w \geq 0.
\]

A collision does not occur, if the value function

\[
d_I(x) := \inf \{g_I(x)^T w \mid G_I(x)^T w = 0, \ w \geq 0\}
\]

is negative for all $I = 1, \ldots, M$. Hence, collisions are avoided by imposing the non-linear and non-differentiable constraints

\[
d_I(x) \leq -\varepsilon, \quad I = 1, \ldots, M,
\]

for some $\varepsilon > 0$. The resulting optimization problem has the advantage of being small compared to the previously discussed problems, but it has the disadvantage of being non-smooth and thus methods for non-smooth optimization problems are required. The investigation of properties of $d_I(x)$ regarding convexity and subdifferentiability and the exploitation in optimization methods is supposed to be investigated in a future activity.

The solutions of the linear programs $\text{LP}_I(x^{(0)})$, $I = 1, \ldots, M$, for a given initial guess $x^{(0)}$ also provide an initial guess for the artificial control variables $w_I$, $I = 1, \ldots, M$. 
REFERENCES


Appendix A. Details of the Robot Model

The configuration of the manipulator robot in three space dimensions is depicted in Figure 11. Herein, \( q = (q_1, q_2, q_3)^T \) denotes the vector of joint angles at the joints of the robot and \( q' = (q'_1, q'_2, q'_3)^T \) denotes the vector of joint angle velocities.

Let the rotation matrices be defined as

\[
S_1(\alpha) = \begin{pmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{pmatrix},
S_2(\beta) = S_3(\beta) = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \beta & -\sin \beta \\
0 & \sin \beta & \cos \beta
\end{pmatrix},
\]
The kinetic energy of the manipulator robot is given by

\[ T(q, q') = \frac{1}{2} \sum_{i=1}^{4} (m_i \| R_i \|^2 + \omega_i^\top J \omega_i), \]

where \( m_i \) are the masses of the links, \( R_i \) are the rotation matrices, \( \omega_i \) are the angular velocities, and \( J \) is the inertia matrix.

The centers of gravity of the three links of the manipulator in the reference system are given by

\[ R_1 = \begin{pmatrix} 0 \\ 0 \\ h_1/2 \end{pmatrix}, \]
\[ R_2 = P_1 + S_{12}(q_1, q_2) \begin{pmatrix} \ell_1 \\ \ell_2/2 \\ 0 \end{pmatrix} = \begin{pmatrix} -b_1 \cos q_1 - \ell_2/2 \sin q_1 \cos q_2 \\ -b_1 \sin q_1 + \ell_2/2 \cos q_1 \cos q_2 \\ h_1 + \ell_1 \sin q_2 + \ell_2/2 \sin(q_2 + q_3) \end{pmatrix}, \]
\[ R_3 = P_2 + S_{123}(q_1, q_2, q_3) \begin{pmatrix} \ell_2/2 \\ 0 \end{pmatrix} = \begin{pmatrix} (b_2 - b_1) \cos q_1 - \sin q_1 (\ell_1 \cos q_2 + \ell_2 \cos(q_2 + q_3)) \\ (b_2 - b_1) \sin q_1 + \cos q_1 (\ell_1 \cos q_2 + \ell_2 \cos(q_2 + q_3)) \\ h_1 + \ell_1 \sin q_2 + \ell_2 \sin(q_2 + q_3) \end{pmatrix}. \]

The center of gravity of the load is given by

\[ R_4 = \begin{pmatrix} (b_2 - b_1) \cos q_1 - \sin q_1 (\ell_1 \cos q_2 + \ell_2 \cos(q_2 + q_3)) \\ (b_2 - b_1) \sin q_1 + \cos q_1 (\ell_1 \cos q_2 + \ell_2 \cos(q_2 + q_3)) \\ h_1 + \ell_1 \sin q_2 + \ell_2 \sin(q_2 + q_3) \end{pmatrix}. \]

Hence, the rotation matrix and the translational vector related to each link of the robot and the load are given by

\[ S^{(1)} = S_1(q_1), \quad r^{(1)} = 0, \]
\[ S^{(2)} = S_{12}(q_1, q_2), \quad r^{(2)} = P_1, \]
\[ S^{(3)} = S_{123}(q_1, q_2, q_3), \quad r^{(3)} = P_2, \]
\[ S^{(4)} = S_{123}(q_1, q_2, q_3), \quad r^{(4)} = R_4. \]

The angular velocities of the respective body coordinate systems with respect to the reference coordinate system expressed in the body reference coordinate system are given by

\[ \omega_1 = \begin{pmatrix} 0 \\ 0 \\ q_1' \cos q_2 \end{pmatrix}, \quad \omega_2 = \begin{pmatrix} q_2' \\ q_1' \sin q_2 \\ q_1' \cos q_2 \end{pmatrix}, \quad \omega_3 = \omega_4 = \begin{pmatrix} q_2' + q_3' \\ q_1' \sin(q_2 + q_3) \\ q_1' \cos(q_2 + q_3) \end{pmatrix}. \]
where $m_i$ and $J_i = \text{diag}(J_{x,i}, J_{y,i}, J_{z,i}), i = 1, 2, 3, 4$, denote the masses of the robot links and the load and the moments of inertia, respectively.

Application of the torques $u_1, u_2, \text{ and } u_3$ at the centers of gravity of the robot links allow to control the robot. The equations of motion are then given by

(10) \[ q'' = M(q)^{-1} (G(q, q') + F(q)), \]

where $M(q) := \nabla^2_{q', q} T(q, q')$ denotes the symmetric and positive definite mass matrix, $G(q, q') := \nabla_q T(q, q') - (\nabla^2_{q', q} T(q, q')) q'$ denotes the generalized Coriolis forces and

\[
F(q) = \begin{pmatrix}
  u_1 \\
  u_2 + u_3 - g\ell_1 \cos q_2 \left( \frac{m_2}{2} + m_3 + m_4 \right) - g\ell_2 \cos(q_2 + q_3) \left( \frac{m_3}{2} + m_4 \right) \\
  u_3 - g\ell_2 \cos(q_3 + q_2) \left( \frac{m_3}{2} + m_4 \right)
\end{pmatrix}
\]

denotes the vector of applied joint torques and gravity forces.