

## OPTIMIZATION-BASED APPROACH TO PATH PLANNING FOR CLOSED CHAIN ROBOT SYSTEMS

WOJCIECH SZYNKIEWICZ \*, JACEK BŁASZCZYK \*\*

\* Institute of Control and Computation Engineering  
Warsaw University of Technology, ul. Nowowiejska 15/19, 00–665 Warsaw, Poland  
e-mail: W.Szynkiewicz@elka.pw.edu.pl

\*\*Research and Academic Computer Network (NASK)  
ul. Wąwózowa 18, 02–796 Warsaw, Poland  
e-mail: jacekb@nask.pl

An application of advanced optimization techniques to solve the path planning problem for closed chain robot systems is proposed. The approach to path planning is formulated as a “quasi-dynamic” NonLinear Programming (NLP) problem with equality and inequality constraints in terms of the joint variables. The essence of the method is to find joint paths which satisfy the given constraints and minimize the proposed performance index. For numerical solution of the NLP problem, the IPOPT solver is used, which implements a nonlinear primal-dual interior-point method, one of the leading techniques for large-scale nonlinear optimization.

**Keywords:** closed chain, path planning, nonlinear optimization.

### 1. Introduction

In our everyday lives we perform many operations in which our two hands cooperate to manipulate diverse objects. Two cooperative hands, if properly utilized, are capable of grasping and manipulating a much larger class of objects, including long, flexible, irregularly shaped, or complex objects (e.g., with internal degrees of freedom). In cooperative manipulation, when multiple robot arms or multi-fingered hands grasp a movable object, a closed kinematic loop (or loops) is formed by the arms or fingers and the object (Asfour *et al.*, 2006; Han and Amato, 2000; Szynkiewicz, 2003; Yakey *et al.*, 2001). There are many other applications such as modular reconfigurable robots, walking machines, and parallel manipulators, in which one or more closed chain linkages exist or can be formed (Merlet, 2000).

It should be noted that closure constraints are common in many domains beyond robotics, for example, in virtual prototyping, computer graphics, and computational chemistry (Fiser *et al.*, 2000; Kallmann *et al.*, 2003). An underlying commonality among such systems is that they have many Degrees Of Freedom (DOFs). In this case, motion planning usually involves high-dimensional confi-

guration spaces and complicated constraints. While closed chain systems can offer advantages over open chains in terms of the rigidity of the whole mechanism, motion planning is complicated by the need to maintain the closed loop structure, described by the closure constraint. For linkages with closed kinematic chains, loop closure constraints restrict the configuration space to a real algebraic variety that is not necessarily parameterizable (Latombe, 1991; LaValle, 2006). In this paper we are focused on path planning for closed chain mechanisms.

Several strategies for path planning for closed chain systems have been proposed in the last years (Cortés *et al.*, 2002; Han and Amato, 2000; Trinkle and Milgram, 2002; Tang *et al.*, 2007; Szynkiewicz and Gosiewski, 1995; Yakey *et al.*, 2001). All existing techniques for solving a motion planning problem for closed linkages have some limitations. In theory, complete algorithms can be applied to systems with holonomic equality constraints such as those imposed by closed kinematic chains (Canny, 1988). Complete, polynomial-time algorithms are limited to some simplified problems, e.g., path planning for planar linkages neglecting collisions with obstacles and with other links (Trinkle and Milgram, 2002). Recently, complete algorithms that take into account point ob-

stacles have been proposed (Liu and Trinkle, 2005). However, computational complexity of exact algorithms still restricts their use in practical problems with complex linkages and work spaces with various obstacles.

The most effective robot path planning strategies today are built upon randomized search techniques. Sample-based algorithms such as Probabilistic Roadmap Methods (PRMs) (Kavraki *et al.*, 1996) and Rapidly-exploring Randomized Trees (RRTs) (Kuffner and LaValle, 2000; Abbasi-Yadkori *et al.*, 2010) have been adapted to solve path planning problems for closed chain linkages (Cortés *et al.*, 2002; Han and Amato, 2000; Yakey *et al.*, 2001; Yerushova and LaValle, 2009).

The general methodology of PRMs and RRTs is to construct a graph (the roadmap) during preprocessing to capture the connectivity of the valid subset of the configuration space and then to query the roadmap to find a path for a given motion planning task. The roadmap becomes suitable for path planning when the following features can be achieved: There is a one-to-one correspondence between components of a graph and components of a free configuration space (*C-free*). Given a point in *space C-free*, it is relatively easy to find a path connecting it to the graph. The main problem is that the probability that a randomly sampled point belongs to the closure set is equal to zero.

It is very difficult to find and connect configurations that satisfy the closure constraints using only randomized techniques. While several strategies use specific heuristics to improve the probability of sampling closed configurations (Cortés *et al.*, 2002), sampling is still very difficult and time consuming for high-dimensional systems. A considerable improvement is obtained by breaking the kinematic loops and employing both forward and inverse kinematics (Han and Amato, 2000). Each loop in the linkage is broken into two sub-chains called respectively the “active chain” and the “passive chain”. For node generation, standard random sampling techniques and forward kinematics are applied to the active chain, and then inverse kinematics are used for the passive chain in order to enforce the closure constraints. However, closed-form inverse kinematics solutions of the passive chain are required for the efficiency of roadmap computation.

The method developed by Cortés *et al.* (2002) is an extension of the approach proposed by Han and Amato (2000). They proposed a simple geometric technique called the Random Loop Generator (RLG) to generate configurations in each single loop. The main goal of the RLG is to generate random configurations for the active chain that have a high probability to be reachable by the passive chain.

More recently, Han *et al.* (2006) proposed a new set of geometric parameters, some inter-joint distances and triangle orientation data, for closed chain linkages such that the problem can be reformulated as a system of linear inequalities. Then, linear programming and other algo-

thms can be used for sampling and local planning. Instead of randomly sampling in the joint angle space to find closed configurations, Tang *et al.* (2007) proposed the so-called Planning with Reachable Distance (PRD). In PRD, the subspace where closed constraints are satisfied is pre-computed and then sampling is performed directly in this subspace. It should be noted that randomized techniques have wildly varying performance and are not complete, they are not guaranteed to find a solution when one exists, nor can they determine that a solution does not exist when that is the case. Some of them achieve a weaker notion of probabilistic completeness, i.e., they find a path with some probability whenever one exists.

Robot motion planning is often viewed as an optimization problem that aims to minimize a given objective function. Numerous approaches have been proposed to solve this problem. A hybrid algorithm for optimizing robot trajectories to achieve minimal motion time and to avoid collisions was described by Zhang and Knoll (1995). Zefran and Kumar (1997) proposed a variational formulation of a general motion planning and numerical algorithm for solving such variational problems. A functional gradient descent technique called CHOMP (Covariant Hamiltonian Optimization for Motion Planning) was used for motion planning for a 7 DOF manipulator (Ratliff *et al.*, 2009). Kanehiro *et al.* (2008) described a local method for collision avoidance between non-strictly convex polyhedra with continuous velocities as an optimization problem with linear inequality constraints.

In this paper we propose an optimization-based approach to path planning for robot systems with closed kinematic chains. Originally, the path planning problem is formulated as a calculus of variation problem involving a conditional extreme. To obtain a numerical solution, this problem is converted to a finite-dimensional mathematical programming problem. To solve it, an advanced optimization method, namely, the interior point IPOPT algorithm, is utilized. In the proposed method only direct kinematics of the robot arms is used. For practical applications, kinematic motion planning is employed more frequently than dynamic motion planning.

Amongst numerical optimization methods for effectively solving large-scale nonlinear problems, at present the most popular are Sequential Quadratic Programming (SQP) algorithms and interior-point methods for nonlinear problems. In terms of computational effectiveness and robustness, interior-point methods are better fitted to large-scale nonlinear optimization than SQP methods. In publications from the nonlinear programming area it is possible to come across a lot of reports comparing numerical performance of solvers based on SQP and interior-point methods for nonlinear problems using the methodology of the performance profiles (Dolan and Moré, 2002). It is worthwhile to mention the works of Benson *et al.* (2002) and Morales *et al.* (2001), comparing SNOPT, filterSQP,

LOQO and KNITRO solvers, along with those by Wächter (2002) as well as Wächter and Biegler (2006), where the IPOPT solver was benchmarked with LOQO and KNITRO solvers. However, Błaszczyk (2007) and Błaszczyk *et al.* (2007) compare the effectiveness of specialized SQP solvers using the problem structure and the IPOPT solver for dynamic optimization problems, for which the solution is obtained as a result of their transformation to NLP problems.

In this work, to solve the path planning problem for closed chain robotic systems, the IPOPT<sup>1</sup> solver was used, which is based on the primal-dual interior-point method for nonlinear problems (Wächter and Biegler, 2006). The outline of this paper is as follows. In Section 2, the configuration space of a closed chain system is defined. The kinematic description of two robot arms handling a single object is given in Section 3. The admissible path planning as a variational problem involving a conditional extreme is formulated in Section 4. The algorithm of solving the NLP problem in a finite-dimensional space is presented in Section 5. A numerical example to demonstrate the effectiveness of the proposed method is given in Section 6. In Section 7 some concluding remarks are presented.

## 2. Configuration space of a closed chain system

Suppose that the world,  $\mathcal{W}$ , is a subset of the 3-dimensional Euclidean space, i.e.,  $\mathcal{W} \subset \mathbb{R}^3$ . Consider a mechanism (linkage)  $\mathcal{M} = (\mathcal{L}, \mathcal{J})$  consisting of links  $L_i \in \mathcal{L}$  (rigid bodies) connected by joints  $J_k \in \mathcal{J}$ , where  $\mathcal{L} = \{L_1, \dots, L_{n_l}\}$  is a finite set of  $n_l$  links and  $\mathcal{J}$  is a set of joints. One can use a standard parameterization technique for  $\mathcal{M}$  (Latombe, 1991). Then the configuration of  $\mathcal{M}$  can be expressed as a vector  $\mathbf{q} = [q_1, \dots, q_n]^T$  of real-valued parameters. Let  $\mathcal{C}$  denote the configuration space (or the C-space) of the mechanism. Let  $\mathcal{M}(\mathbf{q})$  denote the transformation of  $\mathcal{M}$  to the configuration given by  $\mathbf{q}$ .

Feasible configurations for  $\mathcal{M}(\mathbf{q})$  should not involve collision between the robot and workspace obstacles, or self-collision among the links. Let  $\mathcal{B} = (B_1, \dots, B_{n_b}) \subset \mathcal{W}$  denote a set of static obstacles in the workspace. If the obstacle region is denoted as  $\mathcal{C}_{obs} \subset \mathcal{C}$ , then the collision-free (*C-free*) portion of the configuration space can be defined as follows (LaValle, 2006):

$$\mathcal{C}_{free} = \left( \bigcup_{i \in [1, n_b]} \{ \mathbf{q} \in \mathcal{C} \mid (\mathcal{M}(\mathbf{q}) \cap B_i = \emptyset) \} \right) \cup \left( \bigcup_{i, j \in CP} \{ \mathbf{q} \in \mathcal{C} \mid (L_i(\mathbf{q}) \cap L_j(\mathbf{q}) = \emptyset) \} \right), \quad (1)$$

where  $CP$  denotes the set of collision pairs, in which each collision pair  $(i, j) \in CP$  ( $i \neq j$ ) represents a pair of links  $L_i(\mathbf{q})$  and  $L_j(\mathbf{q})$  that are not allowed to be in a configuration  $\mathbf{q}$ .

In general, the closure constraints are holonomic constraints, and can be expressed in the form  $\mathbf{f}(\mathbf{q}) = \mathbf{0}$ . Configurations of a closed chain mechanism belong to the set

$$\mathcal{C}_{clo} = \{ \mathbf{q} \mid \mathbf{q} \in \mathcal{C} \wedge \mathbf{f}(\mathbf{q}) = \{ f_1(\mathbf{q}) = 0, \dots, f_m(\mathbf{q}) = 0 \} \}. \quad (2)$$

If the dimension of  $\mathcal{C}$  is equal to  $n$ , then  $m < n$ , and  $\mathcal{C}_{clo} \subset \mathcal{C}$ . It should be noted that closure constraints can be transformed to the zeros of a system of polynomial equations using a projective transformation (Latombe, 1991). The set of all configurations for linkages that maintains a closed kinematic chain is generally not a manifold. Equation (2) restricts the configuration space to a real algebraic variety,  $\mathcal{C}_{clo}$ , embedded in  $\mathcal{C}$ . However, the algebraic variety  $\mathcal{C}_{clo}$  can be stratified into a collection of manifolds (Latombe, 1991; LaValle, 2006).

Finally, the valid configurations of a closed chain mechanism that satisfy collision constraints belong to the set,  $\mathcal{C}_{val}$ :

$$\mathcal{C}_{val} = \mathcal{C}_{clo} \cap \mathcal{C}_{free}. \quad (3)$$

## 3. Kinematics of two cooperating robot arms

Let us consider a closed chain mechanism consisting of two robot manipulators transferring a single rigid object as shown in Fig. 1. Since the object, the two robot arms and the ground form a closed kinematic chain, the kinematic constraints imposed on the system restrict the reachable locations by the object.

The linkage can be treated as a chain that is composed of two sub-chains (Fig. 1). The configuration of a sub-chain  $i$ ,  $i = 1, 2$  can be specified by its base configuration and its vector of generalized variables  $\mathbf{q}^i \in \mathbb{R}^{n_i}$ . The configuration of the closed chain is described by the vector  $\mathbf{q} = [(\mathbf{q}^1)^T, (\mathbf{q}^2)^T]^T \in \mathbb{R}^n$ , where  $n = n_1 + n_2$ . To specify the position and orientation of coordinate frames shown in Fig. 1, the following  $(4 \times 4)$  homogeneous transformation matrices are defined: the matrix  $T_{wb_i}$ ,  $i = 1, 2$  describes the pose of the  $i$ -th robot base frame  $F_{b_i}$  in the world frame  $F_w$ , the matrix  $T_{b_i e_i}(\mathbf{q}^i)$ ,  $i = 1, 2$  is the forward kinematic transformation of the  $i$ -th sub-chain, which determines the end-effector's frame  $F_{e_i}$  in the  $F_{b_i}$  frame, and the matrix  $T_{e_o}$ ,  $i = 1, 2$  describes the object frame  $F_o$  in the end-effector frame  $F_{e_i}$ .

The closure constraints can be obtained by writing down two homogeneous transformation matrices for a coordinate frame of the object. Each of the matrices corresponds to two different paths to the object (respectively sub-chain 1 or sub-chain 2). The closure constraint can be

<sup>1</sup>An open-source C++ version of IPOPT is available at <http://projects.coin-or.org/Ipopt>.

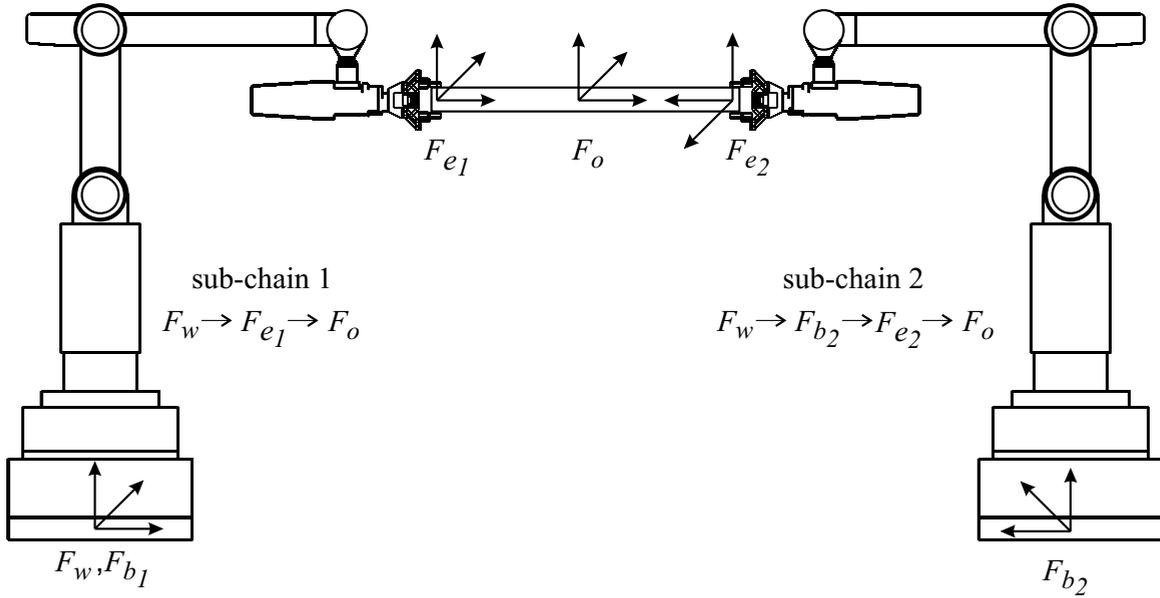


Fig. 1. Closed chain linkage formed by two robot manipulators transferring a common object.

obtained by forcing the frame of the object to be the same, regardless of the path that was chosen. As a consequence, we get the following loop closure constraint:

$$T_{wb_1} T_{b_1e_1}(\mathbf{q}^1) T_{e_1o} = T_{wb_2} T_{b_2e_2}(\mathbf{q}^2) T_{e_2o}. \quad (4)$$

The closure constraint (4) can be reduced to

$$T_{b_1e_1}(\mathbf{q}^1) T_{e_1o} - T_{b_1b_2} T_{b_2e_2}(\mathbf{q}^2) T_{e_2o} = \mathbf{0}, \quad (5)$$

where  $T_{b_1b_2}$  is the transformation from the base of the sub-chain 1 to the base of the sub-chain 2 (Fig. 1). To formulate the path planning problem, we transform the closure constraints described by (5) to the vector form. Using a proper parameterization of the object position and orientation, Eqn. (5) can be rewritten as follows:

$$\mathbf{h}(s, \mathbf{q}) = \mathbf{p}^1(\mathbf{q}^1(s)) - \mathbf{p}^2(\mathbf{q}^2(s)) = \mathbf{0}, \quad \forall s \in [0, 1], \quad (6)$$

where the vector  $\mathbf{p}^i(\mathbf{q}^i(s)) \in \mathbb{R}^6$ ,  $i = 1, 2$ , describes the pose of the object frame  $F_o$  with respect to the world frame  $F_w$  expressed as a function of the generalized variables  $\mathbf{q}^i$ . Mechanical constraints due to mobility limitations of the joints and constraints caused by workspace occupancy conflicts between the object and the robot arms can be described as

$$\mathbf{g}(s, \mathbf{q}) \leq \mathbf{0}. \quad (7)$$

Due to the constraints imposed on the overall system, the problem is to develop an admissible path planning method.

#### 4. Path planning

The path planning problem for the closed chain system shown in Fig. 1 is formulated in terms of the object to be

manipulated. The path planning problem is described as follows:

*Problem statement.* Given a start configuration  $\mathbf{q}_s \in \mathcal{C}_{val}$ , which corresponds to the initial object pose  $\mathbf{p}(0)$ , and a final configuration  $\mathbf{q}_f \in \mathcal{C}_{val}$ , which corresponds to the final object pose  $\mathbf{p}(1)$ , find a continuous joint path  $\mathbf{q}(s) : [0, 1] \rightarrow \mathcal{C}_{val}$  such that  $\mathbf{q}(0) = \mathbf{q}_s$  and  $\mathbf{q}(1) = \mathbf{q}_f$ , where parameter  $s \in [0, 1]$  is the normalized arc length of the spatial curve traversed by the object.

For a path to exist between  $\mathbf{q}_s$  and  $\mathbf{q}_f$ , it will be necessary for both configurations to be contained within the same connected component of  $\mathcal{C}_{val}$ . What makes this class of path planning problems difficult, even though collisions are ignored is the complexity of the valid portion of the C-space,  $\mathcal{C}_{clo}$ , which, as mentioned before, is a real algebraic variety of co-dimension 3 (for spatial closed chains) that is not necessarily parameterizable (LaValle, 2006). However, the dimension of this variety is significantly smaller than that of the ambient space (Latombe, 1991). In such a situation it is of great interest to develop motion planning algorithms which take advantage of this fact and whose complexity reflects the dimension of this variety rather than that of the ambient space.

#### 4.1. Variational formulation of the admissible path planning problem.

As shown by Szykiewicz and Gosiewski (1995), a path planning problem for closed-chain systems can be formulated as a variational problem involving a conditional extreme. We consider the functional in a general form,

$$I[s, \mathbf{q}] = \int_0^1 F(s, \mathbf{q}(s)) ds, \quad (8)$$

subject to

$$\mathbf{q} \in \mathcal{Y} \triangleq \{ \mathbf{q} \in \mathbf{C}^2([0, 1]) : \mathbf{h}(s, \mathbf{q}) = \mathbf{0} \wedge \mathbf{g}(s, \mathbf{q}) \leq \mathbf{0} \wedge \mathbf{q}(0) = \mathbf{q}_s, \mathbf{q}(1) = \mathbf{q}_f \}. \quad (9)$$

Let  $\mathbf{p}^d(\cdot)$  denote a desired path of the object in the task space connecting the initial pose  $\mathbf{p}_s$  and the final pose  $\mathbf{p}_f$ . The desired path is arbitrarily chosen without regard to whether it is admissible. We minimize the functional  $I[\mathbf{q}(\cdot)]$  defined as

$$I[\mathbf{q}(\cdot)] = \int_0^1 (\mathbf{p}^i(\mathbf{q}^i(s)) - \mathbf{p}^d(s))^T W (\mathbf{p}^i(\mathbf{q}^i(s)) - \mathbf{p}^d(s)) ds, \quad (10)$$

where  $\mathbf{p}^i(\cdot)$ ,  $i = 1, 2$  is the current path of the object. The weighting matrix  $W = W^T \geq 0$  is a  $(6 \times 6)$  symmetric positive semi-definite matrix that should be selected appropriately for the task to be performed.

The admissible object path is the closest to the desired path in the sense that the functional  $I$  defined by (10) is minimized with respect to  $\mathbf{q}$  and subject to constraints given by (9). To obtain a numerical solution, the infinite-dimensional optimization problem is approximated by a finite-dimensional problem of mathematical programming.

**4.2. Finite dimensional approximation.** The proposed approach is based on a modified *Ritz method* (Daniel, 1971), one of the direct methods of the calculus of variations. The basic idea of the Ritz method is that the values of a functional (10) are considered not on arbitrary admissible curves of a given variational problem but only on all possible linear combinations,

$$\mathbf{q}^{(N)}(\cdot) = \sum_{j=0}^N \mathbf{c}_j^{(N)} \varphi_j, \quad \dim \mathbf{q} = \dim \mathbf{c}_j^{(N)}, \quad (11)$$

of the first  $N + 1$  functions of some chosen sequence of functions,  $\{\varphi_j\}_{j=0}^\infty$ , where  $\mathbf{c}_j$  denotes the undetermined parameters termed Ritz coefficients. The functions  $\mathbf{q}^{(N)}$  given in (11) must be admissible for the functional (10).

A convergent Ritz approximation requires the following (Daniel, 1971):

1.  $\varphi_j$  must satisfy the specified essential boundary conditions.
2.  $\varphi_j$  must be continuous and smooth enough, as required by the variational statement being used.
3. The set  $\{\varphi_j\}$  must be linearly independent and complete.

If the basis functions satisfy the requirements, the assumed approximation  $\mathbf{q}^{(N)}$  normally converges to the actual solution  $\bar{\mathbf{q}}$  when  $N \rightarrow \infty$ .

To formulate a finite-dimensional nonlinear optimization problem, the basis is defined as a sequence  $\{\varphi_j\}_{j=0}^N$ , where  $\varphi_j$ ,  $j = 0, 1, \dots, N$  is a set of appropriately chosen basis functions.

The quality of the obtained solution depends on a proper choice of the basis functions. In our approach, we choose cubic B-splines (de Boor, 1978) as the basis functions.

Let the interval  $[0, 1] \subset \mathbb{R}^1$  be divided into  $N$  equal segments  $\Delta s = s_{j+1} - s_j = 1/N$  for  $s_j$ ,  $j = 0, 1, \dots, N - 1$ , where  $s_j$  are values such that  $0 = s_0 < s_1 < \dots < s_N = 1$ .

Denote by  $\mathbb{B}^2$  the cubic B-spline space. The basis in the space  $\mathbb{B}^2$  can be defined using the following formula (de Boor, 1978):

$$b_{j,0}(s) = \begin{cases} 1 & \text{if } s \in [s_j, s_{j+1}] \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The (cubic) B-spline function on the interval  $[s_j, s_{j+4}]$  is defined as

$$b_{j,3}(s) = \frac{s - s_j}{s_{j+3} - s_j} b_{j,2}(s) + \frac{s_{j+4} - s}{s_{j+4} - s_{j+1}} b_{j+1,2}(s). \quad (13)$$

The admissible path finding problem is reduced to minimizing a function of  $(N + 3) \times n$  variables. Without loss of generality, the weight matrix can be defined as  $W = \text{diag}(w_1, \dots, w_6)$ . Now, the functional (10) can be expressed as a function of the coefficients  $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T]^T$  ( $\mathbf{c}_i \in \mathbb{R}^{n_i \times (N+3)}$ ,  $i = 1, 2$ ) of the path  $\mathbf{q}(\cdot)$  in the basis  $\{\varphi_j\}_{j=0}^{N+2}$ ,

$$\min_{\mathbf{c}} \sum_{j=0}^{N-1} \int_{s_j}^{s_{j+1}} \sum_{k=1}^6 w_k \left( p_{1k} \left( \sum_{i=j}^{j+3} \mathbf{c}_{1i} \varphi_i(s) \right) - p_k^d(s) \right)^2 ds. \quad (14)$$

The function (14) is minimized with respect to the vector variable  $\mathbf{c}$ , and subject to

$$\mathbf{c} \in \mathcal{Y}_N = \{ \mathbf{c} \in \mathbb{R}^{n \times (N+3)} \mid \mathbf{h}(s_j, \mathbf{c}) = \mathbf{0} \wedge \mathbf{g}(s_j, \mathbf{c}) \leq \mathbf{0}, j = 0, \dots, N \}. \quad (15)$$

The admissible path planning problem was reduced to a finite-dimensional nonlinear optimization problem. It can be solved by means of appropriate, efficient methods of mathematical programming, particularly interior-point or barrier algorithms for large-scale nonlinear programming.

## 5. Interior-point algorithm for NLP problems

Interior-point methods for nonlinear programming, also called barrier methods, rose from the need for effective

solving of large-scale optimization problems. In particular, for NLP problems with a large number of inequality constraints, these methods offer a serious alternative to active-set strategies. Within the last 15 years, researchers have led to a better understanding of the convergence of interior-point methods and have also developed effective computational algorithms with desirable global and local convergence properties.

The term *interior-point method* was used for the first time by Fiacco and McCormick (1968) for any algorithm that was designed for the calculation of a local minimum of an NLP problem by the solution of a determined sequence of unconstrained minimization problems. Such a definition evolved to the form in which we think of the IP method as of any algorithm for solving a set of optimization problems associated with a decreasing value of the multiplier  $\mu$ , to find local solutions lying in the interior of the feasibility set determined by nonlinear constraints of the NLP problem.

To allow convergence from “bad” starting points for interior-point methods in both trust region and line-search versions, researchers developed exact penalty merit functions that ensure progress toward the solution (Byrd *et al.*, 2000; Tits *et al.*, 2002). On the other hand, Fletcher and Leyffer (2002) recently proposed filter methods as an alternative to merit functions which guarantee the global convergence for nonlinear programming algorithms. They are based on the idea of the approval of trial points generated by the optimization algorithm in the case when they improve the value of the objective function *or* improve the value of a constraint violation, instead of a combination of those two measures defined by a merit function.

More recently, this filter technique has been adapted to barrier methods. Ulbrich *et al.* (2004) consider a trust-region filter method, in which the consequent iterations of the solution are accepted on the basis of the norm of optimality conditions. Also, Benson *et al.* (2001) proposed several heuristics based on the concept of filter methods, for which the efficiency improvement was obtained as compared with their previous experience with merit functions. Finally, Wächter and Biegler (2005) provided global convergence analysis of an interior-point algorithm with a filter line-search.

Interior-point methods for NLP problems were implemented within many optimization solvers, such as LO-QO (Vanderbei and Shanno, 1997), KNITRO (Byrd *et al.*, 1999; Waltz and Plantenga, 2006) or IPOPT (Wächter, 2002; Wächter and Biegler, 2006). In numerical tests these solvers proved to be quite effective and robust for many large-scale NLP problems.

**5.1. IPOPT solver.** To solve the path planning problem for closed chain robot systems, we used a primal-dual interior-point algorithm with line-search minimization based on the filter method, employed in the implementation

of the IPOPT solver. A formal description and analysis of the filter line-search procedure implemented in the IPOPT solver can be found in the work of Wächter and Biegler (2005). In comparison with traditional line-search algorithms, such as a single merit function technique, the filter method is usually less conservative and makes it possible to take a larger step size. Moreover, the protection in the form of a restoration phase makes the filter algorithm resistant to unnecessary errors, such as those presented by Wächter and Biegler (2000).

The computationally most expensive part of the optimization algorithm implemented in the IPOPT solver (not including computations of the objective function, constraints and their derivatives) is the solution of the linear system of equations, which is most often of high order and has a sparse structure. For its factorization and solution, IPOPT uses external sparse direct linear solvers, such as MA27 (default option), MA57, WSMP, PARDISO and MUMPS.

## 6. Numerical example

Let us consider a system consisting of two Asea IRb-6 robot arms (robots have only 5 DOF each) manipulating a rigid beam as shown in Fig. 1 (Haegele *et al.*, 2008).

The parameters of the Denavit–Hartenberg notation and joint ranges of the IRb-6 manipulator are listed in Table 1. The forward kinematics for the IRb-6 robot arm are

Table 1. Denavit–Hartenberg parameters and joint ranges of the Asea IRb-6 robot manipulator.

Joint $i$	$\alpha_{i-1}$ [rad]	$a_{i-1}$ [m]	$d_i$ [m]	$[\theta_i^{\min}, \theta_i^{\max}]$ [rad]
1	0	0	0.7	$[-\frac{17}{18}\pi, \frac{17}{18}\pi]$
2	$-\pi/2$	0	0	$[-\frac{13}{18}\pi, -\frac{5}{18}\pi]$
3	0	0.45	0	$[-\frac{25}{180}\pi, \frac{4}{18}\pi]$
4	0	0.67	0	$[-\frac{1}{2}\pi, +\frac{1}{2}\pi]$
5	$-\pi/2$	0	0	$[-\pi, +\pi]$
6			0.1	

given by a  $(4 \times 4)$  homogenous transformation matrix (6),

$$T_{b_i e_i}(\mathbf{q}^i) = \begin{bmatrix} c_{i1}s_{i4}c_{i5} + s_{i1}s_{i5} & -c_{i1}s_{i4}s_{i5} + s_{i1}c_{i5} & c_{i1}c_{i4} & c_{i1}c_{i4} \\ s_{i1}s_{i4}c_{i5} - c_{i1}s_{i5} & -s_{i1}s_{i4}s_{i5} - c_{i1}c_{i5} & s_{i1}c_{i4} & s_{i1}c_{i4} \\ c_{i4}c_{i5} & -c_{i4}s_{i5} & -s_{i4} & -s_{i4} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{i1}(d_{i6}c_{i4} + a_{i3}c_{i3} + a_{i2}c_{i2}) \\ s_{i1}(d_{i6}c_{i4} + a_{i3}c_{i3} + a_{i2}c_{i2}) \\ d_{i1} - d_{i6}s_{i4} - a_{i3}s_{i3} - a_{i2}s_{i2} \\ 1 \end{bmatrix},$$

where  $s_{ij} = \sin(\theta_{ij})$  and  $c_{ij} = \cos(\theta_{ij})$ ,  $i = 1, 2$ ,  $j = 1, \dots, 5$ . Joint angles  $\theta_{ij}$  represent configuration variables, other parameters are given in Table 1.

The pose of the base of the robot manipulator 2 with respect to the world coordinate frame  $F_w$  is given by

$$T_{wb_2} = \begin{bmatrix} -1 & 0 & 0 & l_0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where  $l_0 = 1.8$  [m] is the distance between origins of the coordinate frames  $F_{b_1}$  and  $F_{b_2}$ .

The pose of the object frame with respect to the end-effector frames  $F_{e_i}$  is given by the following matrices:

$$T_{e_1o} = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{e_2o} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where  $l_1 = l_2 = 0.2$  [m] denote the distances between origins of the coordinate frame  $F_o$  and  $F_{e_1}$  and  $F_{e_2}$ , respectively. The desired object path  $\mathbf{p}^d(\cdot)$  in the coordinate frame  $F_w$  is defined as the straight line path

$$\mathbf{p}^d(s) = \mathbf{p}^d(0)(1-s) + \mathbf{p}^d(1)s, \quad s \in [0, 1].$$

The initial and final object locations in the  $F_w$  reference frame are given in Table 2. Object orientation is parame-

Table 2. Initial and final object locations in the reference frame.

Coordinate	Object pose	
	Initial	Final
x [m]	0.90	0.95
y [m]	0	0.30
z [m]	0.90	1.00
$\phi$ [rad]	0	0.3059
$\theta$ [rad]	0	$\pi/2$
$\psi$ [rad]	0	0

terized by three Z-Y-Z Euler angles, respectively  $\phi$ ,  $\theta$ ,  $\psi$ . The corresponding initial and final configuration of both robot manipulators is given in Table 3. We assume that the initial and final object poses and corresponding configurations of both robot arms are feasible. (This means that the initial and final configuration belong to the set of valid configurations,  $\mathbf{q}_s, \mathbf{q}_f \in \mathcal{C}_{val}$ .)

The discussed NLP problem for the system of two cooperating robots was implemented in the C++ programming language using classes and inheritance mechanisms. In addition, for handling vector-matrix operations,

Table 3. Initial and final joint values for two robot arms.

Joint variable	Robot 1		Robot 2	
	Initial	Final	Initial	Final
$q_1$ [rad]	0	0.3059	0	-0.3393
$q_2$ [rad]	-1.4674	-0.9087	-1.6193	-0.9039
$q_3$ [rad]	0.2221	0.3903	0.3815	-0.3766
$q_4$ [rad]	$\pi/2$	0	0	$\pi/2$
$q_5$ [rad]	0	0	0	-0.6451

the uBLAS library was used. Specifically, classes for fixed size vectors and matrices were implemented, and also a class for handling homogeneous transformation matrices, based on the available implementation in the MRROC++ robot programming framework (Zieliński and Winiarski, 2010).

MRROC++ contains, among other things, methods for the conversion of homogeneous matrices to Cartesian coordinates and Z-Y-Z or Z-Y-X Euler angles. To solve the resulting NLP problem, the IPOPT solver was used, configured to employ second order derivatives of the objective function and constraints, which had considerable influence on convergence to the local optimum. The IPOPT solver was configured to use the MA57 matrix solver from the Harwell Subroutine Library (HSL). Moreover, a few values of IPOPT parameters were adjusted, which had crucial influence on the convergence.

As the starting point for the IPOPT solver, the values of  $c$ -coefficients were used, obtained from the solution of an auxiliary linear system of equations for a previously computed permissible initial path. The computation of such a path is based on iteratively searching for successive points of the path, solving the NLP problem for a single point, rather than for the entire path. Of course, there is no guarantee that the resulting sequence of points will yield a continuous initial path, but it is a set of acceptable points. In order to avoid computation and programming of complex analytical formulas for the first and the second function derivatives, we used the automatic differentiation library CppAD<sup>2</sup> (Bell and Burke, 2008), which has support for the IPOPT solver.

In the numerical example considered we chose the cubic B-splines as the basis functions. The interval  $[0, 1]$  was partitioned into  $N = 50$  even subintervals of  $\Delta s = 0.02$  length each. Since the initial and final points of the robot paths are fixed, it was required to introduce additional equality constraints to the NLP problem. The convergence behavior of the IPOPT algorithm depends on the form of the objective function and constraints. In the case considered, the objective function is nonconvex and equality constraints are nonlinear. Therefore, IPOPT might converge to a local optimum which is not a global solution. Ho-

<sup>2</sup>An open-source C++ code of the CppAD library is available at <http://projects.coin-or.org/CppAD>.

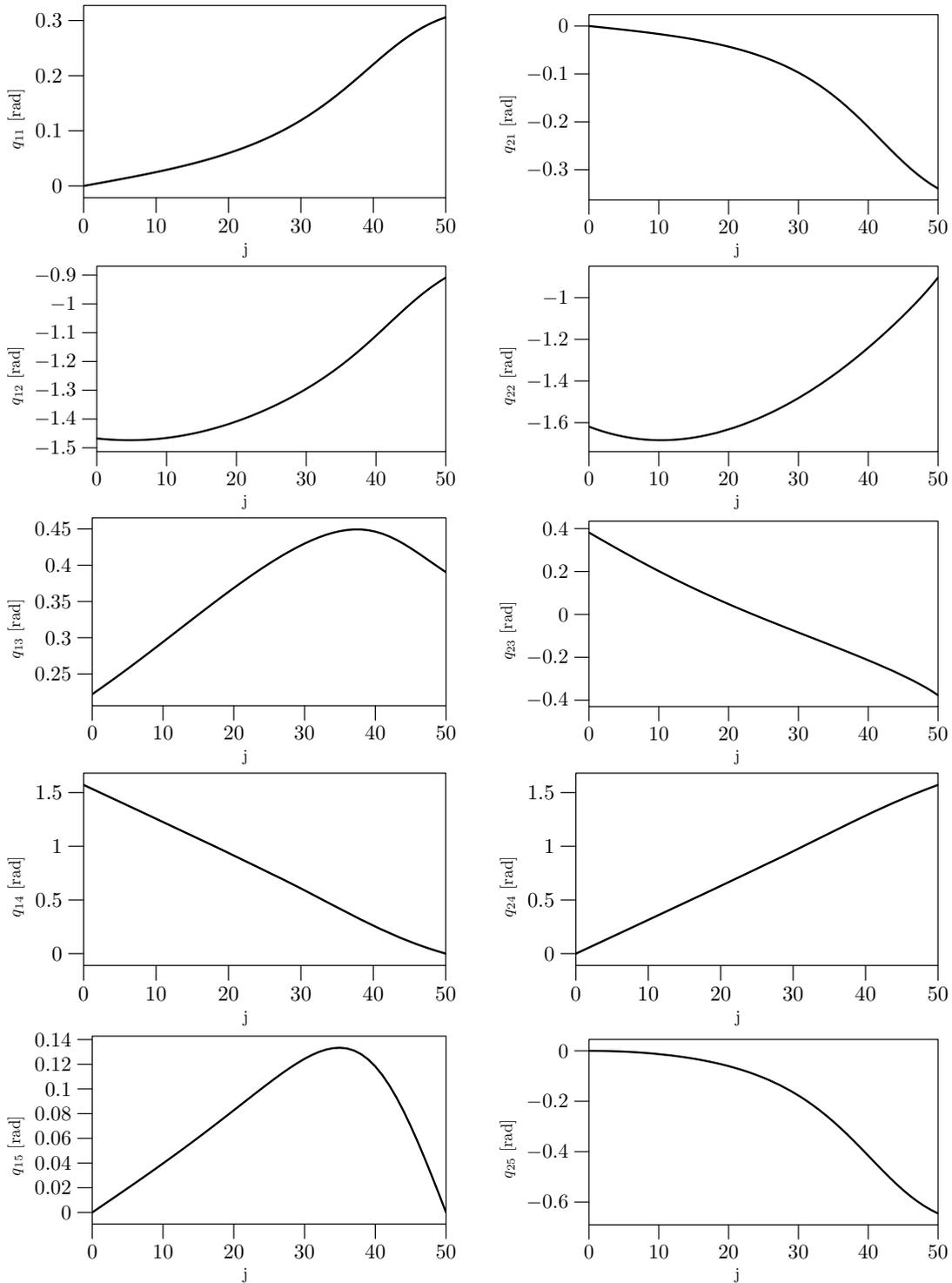


Fig. 2. Paths in joint coordinates of both robot manipulators.

wever, from a practical point of view, it is more important to obtain a feasible solution in a relatively short time than spending a lot of time searching for a global solution. The tolerance of equality constraints in the NLP problem for the considered case was about  $10^{-10}$ . The constraint vio-

lation is the norm of the constraint equations (which are supposed to be zero at a feasible point).

Numerical results obtained by solving the optimization problem are presented in Figs. 2–4 and Table 4. In Fig. 2, joint paths of both robot manipulators are shown.

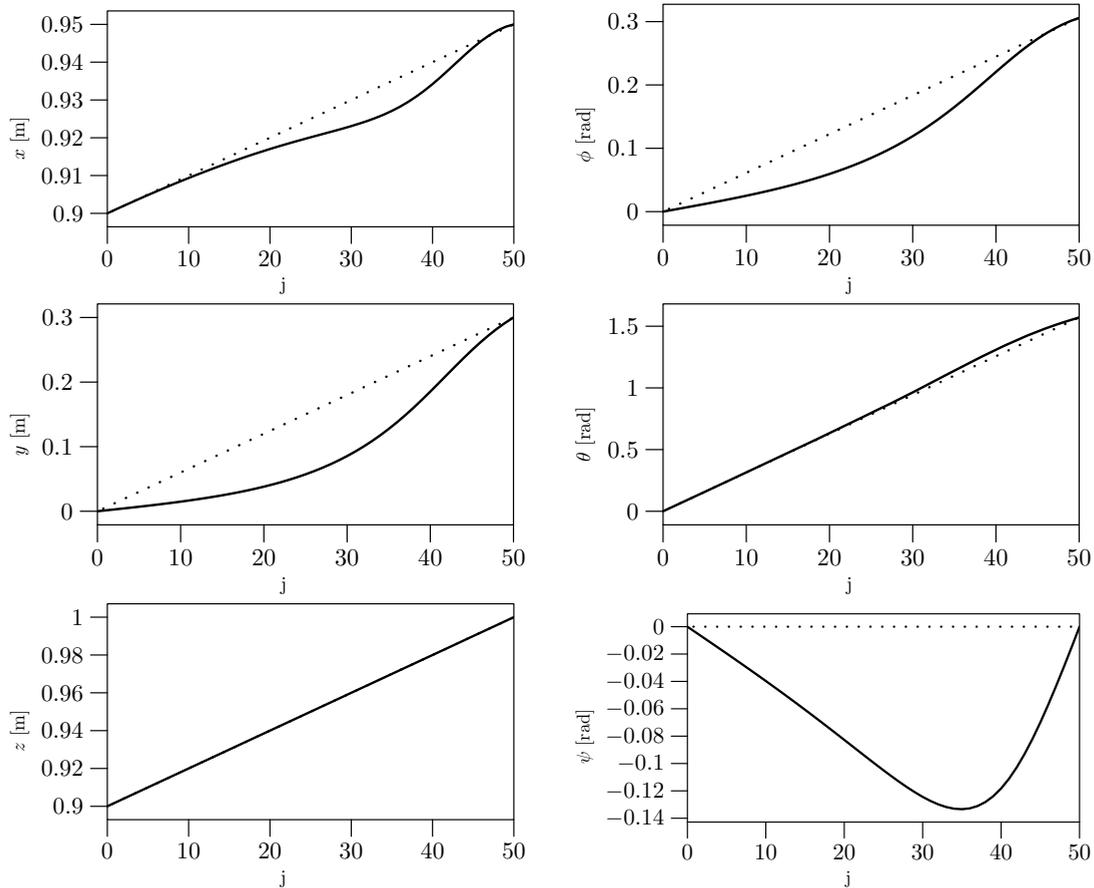


Fig. 3. Desired (dotted line) and actual (solid line) object paths.

The joint paths are always continuous and smooth enough because they are described as a linear combination of the cubic B-spline basis functions. Moreover, as a B-spline curve the path can be locally modified without affecting the whole curve. In Fig. 3 we show Cartesian coordinates  $(x, y, z)$  of the path traversed by the origin of the local frame  $F_o$  associated with the object, and Z-Y-Z Euler angles  $(\phi, \theta$  and  $\psi$ , respectively) representing the orientation of the object with respect to the reference frame,  $F_w$ .

The object path was obtained by solving forward kinematics. The desired object path is drawn with dotted lines, whereas the actual path with the solid one. It should be noted again that the desired object path is not necessarily admissible. It might happen that the object cannot follow this path without violating the constraints. However, the actual path is the “closest” to the desired one in the sense that all constraints are satisfied and the cost function is minimized. In Fig. 4 the object path and orientation in 3D space are shown. Translation motion is a curve in  $\mathbb{R}^3$ , so it can be naturally formulated as a 3D cubic B-spline. Rotation motion is also a 3D cubic spline in 3D “space” of Euler angles. The combination of translation and rotation curves constitutes the motion spline curve.

Comparing our approach to modern sampling based methods, it should be noted that the paradigm underlying sampling-based planners does not by itself enforce any form of solution optimization. Searching for a feasible path may often result in solutions of unpredictable length, with superfluous motions. The resulting path may contain unnecessary turns, or the velocities at the vertices may change arbitrarily. In contrast, paths produced by our algorithm are smooth and “close” do the desired ones, which can be chosen according to the needs. Moreover, as mentioned before, it is very difficult to find and connect configurations that satisfy the closure constraints using only randomized techniques. Combinatorial approaches to motion planning find paths through the continuous configuration space without resorting to approximations. These approaches construct explicit geometrical and topological representations of the closure set, but are usually inefficient in practice. However, due to the difficulty of computing the exact representation of the closure constraints, most implementations of these algorithms are limited to robots with few DOFs or special shapes. Moreover, in our approach there is no need to solve the inverse kinematics problem, which can be difficult and time consuming for mechanisms with many DOFs and many

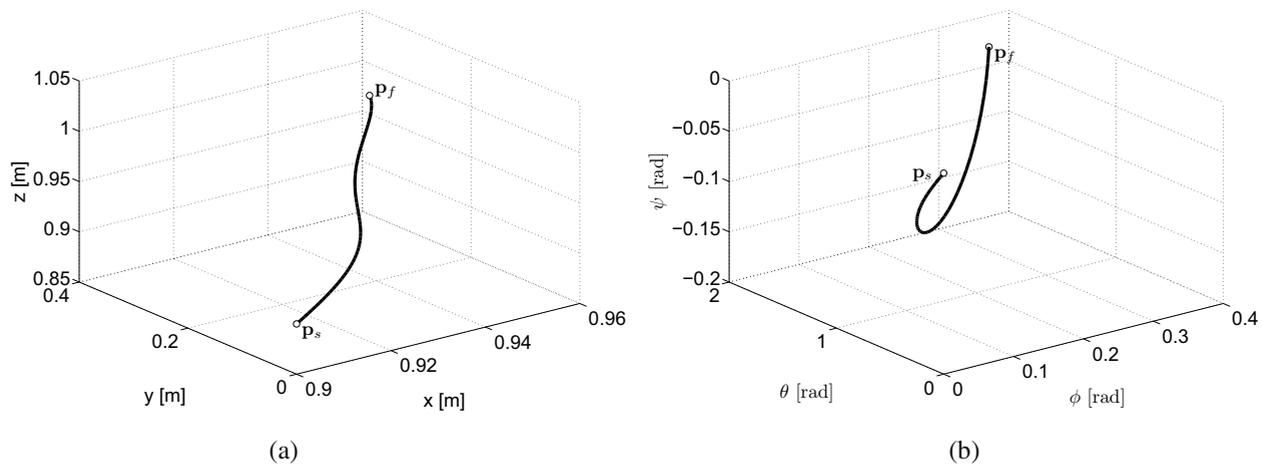


Fig. 4. 3D object paths: position (a), orientation (b).

Table 4. Computational results for the numerical example with  $N$  increasing from 25 to 800 segments ( $n, m_e, m$  stand for the numbers of variables, equality and inequality constraints in NLP problems, respectively). The number of iterations and CPU times (in seconds) are listed for the IPOPT solver. The times are taken using a PC with an Intel Q6600 processor with a 2.4 GHz clock. The parameters  $e_{p,max}$  and  $e_{o,max}$  are the accuracies of the calculated paths for position and orientation, respectively. The SP flag determines whether or not a starting point for the IPOPT solver was generated before optimization.

$N$	$n$	$m_e$	$m$	SP	Iter	Time [s]	$e_{p,max}$ [m]	$e_{o,max}$ [rad]
25	280	144	280	1	13	0.6	1.19e-08	9.77e-11
				0	18	0.9	1.61e-08	1.31e-10
50	530	294	530	1	15	2.5	6.09e-10	4.43e-12
				0	21	3.3	6.13e-10	4.46e-12
100	1030	594	1030	1	19	10.2	6.16e-11	2.27e-12
				0	-	-	-	-
200	2030	1194	2030	1	22	43.4	2.67e-10	4.78e-11
				0	25	50.1	2.68e-10	4.79e-11
400	4030	2394	4030	1	24	193.3	2.25e-10	8.13e-11
				0	26	219.6	2.25e-10	8.13e-11
800	8030	4794	8030	1	24	779.2	1.31e-08	4.67e-09
				0	24	820.1	1.31e-08	4.62e-09

closed chains. Also, additional constraints like task constraints can be easily included in our framework.

### 7. Conclusion

This paper presents an optimization-based approach to path planning for closed chain robotic systems. The path planning problem was formulated as a functional minimization problem with equality and inequality constraints in terms of the joint variables. It was solved numerically. The obtained joint and object paths are always continuous and smooth enough due to cubic B-spline approximation. All kinematic constraints imposed on the two robot arms handling a single rigid object were satisfied. A variational approach to path planning is suitable not only for simple linkages with a single loop and a few links, but also

for complex mechanisms with multiple closed chains and many obstacles in the workspace. It may also be directly applicable to kinematically redundant robot systems. The effectiveness and accuracy depend on the resolution of discretization of the configuration space and the quality of nonlinear programming techniques used to solve the problem.

### Acknowledgment

The authors gratefully acknowledge the support of the Polish Ministry of Science and Higher Education within the grants N N514 1287 33 and N N514 4169 34.

### References

Abbasi-Yadkori, Y., Modayil, J. and Szepesvari, C. (2010). Extending rapidly-exploring random trees for asymptoti-

- cally optimal anytime motion planning, *IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan*, pp. 127–132.
- Asfour, T., Gyarfas, F., Azad, P. and Dillmann, R. (2006). Imitation learning of dual-arm manipulation tasks in humanoid robots, *IEEE/RAS International Conference on Humanoid Robots (Humanoids 2006), Genoa, Italy*, pp. 40–47.
- Bell, B.M. and Burke, J.V. (2008). Algorithmic differentiation of implicit functions and optimal values, in C.H. Bischof, H.M. Bücker, P.D. Hovland, U. Naumann and J. Utke (Eds.), *Advances in Automatic Differentiation*, Springer, Berlin/Heidelberg, pp. 67–77.
- Benson, H.Y., Shanno, D.F. and Vanderbei, R.J. (2001). Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions, *Technical Report ORFE-00-06*, Operations Research and Financial Engineering, Princeton University, Princeton, NJ.
- Benson, H.Y., Shanno, D.F. and Vanderbei, R.J. (2002). A comparative study of large-scale nonlinear optimization algorithms, *Technical Report ORFE-01-04*, Operations Research and Financial Engineering, Princeton University, Princeton, NJ.
- Błaszczak, J., Karbowski, A. and Malinowski, K. (2007). Object library of algorithms for dynamic optimization problems: Benchmarking SQP and nonlinear interior point methods, *International Journal of Applied Mathematics and Computer Science* 17(4): 515–537, DOI: 10.2478/v10006-007-0043-y.
- Błaszczak, J.P. (2007). *Object Library of Algorithms for Dynamic Optimization: A Study on Effectiveness of Sequential Quadratic Programming and Nonlinear Interior Point Methods*, Ph.D. thesis, Warsaw University of Technology, Warsaw, (in Polish).
- Byrd, R.H., Gilbert, J.C. and Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming* 89(1): 149–185.
- Byrd, R.H., Hribar, M.E. and Nocedal, J. (1999). An interior point algorithm for large scale nonlinear programming, *SIAM Journal on Optimization* 9(4): 877–900.
- Canny, J. (1988). *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA.
- Cortés, J., Siméon, T. and Laumond, J.-P. (2002). A random loop generator for planning the motions of closed kinematic chains using PRM methods, *IEEE International Conference on Robotics and Automation ICRA, Washington, DC, USA*, pp. 2141–2146.
- Daniel, J. (1971). *Approximate Minimisation of Functionals*, Prentice Hall, Englewood Cliffs, NJ.
- de Boor, C. (1978). *Practical Guide to Splines*, Springer, New York, NY/Heidelberg.
- Dolan, E.D. and Moré, J.J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming* 91(2): 201–213.
- Fiacco, A.V. and McCormick, G.P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, NY/London.
- Fiser, A., Do, R. and Sali, A. (2000). Modeling of loops in protein structure, *Protein Science* 9(9): 1753–1773.
- Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function, *Mathematical Programming* 91(2): 239–269.
- Haeghele, M., Nilsson, K. and Pires, J.N. (2008). *Springer Handbook of Robotics*, Springer, Berlin/Heidelberg.
- Han, L. and Amato, N. (2000). A kinematics-based probabilistic roadmap method for closed chain systems, *Workshop on Algorithmic Foundations of Robotics, Dartmouth, Hanover, NH, USA*, pp. 233–245.
- Han, L., Rudolph, L., Blumenthal, J. and Valodzin, I. (2006). Stratified deformation space and path planning for a planar closed chain with revolute joints, in S. Akella, N. Amato, W. Huang and B. Mishra (Eds.), *International Workshop on Algorithmic Foundations of Robotics WAFR*, Springer Tracts in Advanced Robotics, Vol. 47, Springer, New York, NY, pp. 235–250.
- Kallmann, M., Aubel, A., Abaci, T. and Thalmann, D. (2003). Planning collision-free reaching motions for interactive object manipulation and grasping, *Eurographics* 22(3): 313–322.
- Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E. and Laumond, J.-P. (2008). A local collision avoidance method for non-strictly convex polyhedra, *Proceedings of Robotics: Science and Systems (RSS) IV, Zurich, Switzerland*, pp. 151–158.
- Kavraki, L.E., Svestka, P., Latombe, J.-C. and Overmars, M.H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation* 12(4): 566–580.
- Kuffner, J.J. and LaValle, S.M. (2000). RRT-connect: An efficient approach to single-query path planning, *IEEE International Conference on Robotics and Automation, San Francisco, CA, USA*, pp. 995–1001.
- Latombe, J.-C. (1991). *Robot Motion Planning*, Kluwer, Boston, MA.
- LaValle, S. (2006). *Planning Algorithms*, Cambridge University Press, Cambridge.
- Liu, G. and Trinkle, J. (2005). Complete path planning for planar closed chains among point obstacles, *Proceedings of Robotics: Science and Systems (RSS) I, Cambridge, MA, USA*, pp. 33–40.
- Merlet, J. (2000). *Parallel Robots*, Kluwer, Dordrecht.
- Morales, J.L., Nocedal, J., Waltz, R.A., Liu, G. and Goux, J.-P. (2001). Assessing the potential of interior methods for nonlinear optimization, *Technical Report OTC 2001/4*, Optimization Technology Center, Northwestern University, Evanston, IL.
- Ratliff, N., Zucker, M., Bagnell, J.A. and Srinivasa, S. (2009). CHOMP: Gradient optimization techniques for efficient motion planning, *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pp. 489–494.

- Szykiewicz, W. (2003). Motion planning for multi-robot systems with closed kinematic chains, *9th IEEE International Conference on Methods and Models in Automation and Robotics, Międzyzdroje, Poland*, pp. 779–786.
- Szykiewicz, W. and Gosiewski, A. (1995). Motion space analysis and trajectory planning for dual-arm system, *2nd International Symposium on Methods and Models in Automation and Robotics, Międzyzdroje, Poland*, pp. 503–510.
- Tang, X., Thomas, S. and Amato, N. (2007). Planning with reachable distances: Fast enforcement of closure constraints, *IEEE International Conference on Robotics and Automation ICRA, Rome, Italy*, pp. 2694–2699.
- Tits, A.L., Wächter, A., Bakhtiari, S., Urban, T.J. and Lawrence, C. (2002). A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties, *Technical Report TR 2002-29*, Institute for Systems Research, University of Maryland, College Park, MD.
- Trinkle, J. and Milgram, R. (2002). Complete path planning for closed kinematic chains with spherical joints, *International Journal of Robotics Research* **21**(9): 773–789.
- Ulbrich, M., Ulbrich, S. and Vicente, L.N. (2004). A globally convergent primal-dual interior-point filter method for nonlinear programming, *Mathematical Programming* **100**(2): 379–410.
- Vanderbei, R.J. and Shanno, D.F. (1997). An interior-point algorithm for non-convex nonlinear programming, *Technical Report SOR-97-21*, Statistics and Operations Research, Princeton University, Princeton, NJ.
- Wächter, A. (2002). *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA.
- Wächter, A. and Biegler, L.T. (2000). Failure of global convergence for a class of interior point methods for nonlinear programming, *Mathematical Programming* **88**(3): 565–574.
- Wächter, A. and Biegler, L.T. (2005). Line search filter methods for nonlinear programming: Motivation and global convergence, *SIAM Journal on Optimization* **16**(1): 1–31.
- Wächter, A. and Biegler, L.T. (2006). On the implementation of a primal-dual interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* **106**(1): 25–57.
- Waltz, R.A. and Plantenga, T. (2006). *KNITRO 5.0 User's Manual*, Ziena Optimization, Inc., <http://www.ziena.com/docs/knitroman.pdf>.
- Yakey, J., LaValle, S. and Kavraki, L. (2001). Randomized path planning for linkages with closed kinematic chains, *IEEE Transactions on Robotics and Automation* **17**(6): 951–958.
- Yershova, A. and LaValle, S. (2009). Motion planning for highly constrained spaces, in K.R. Kozłowski (Ed.) *Robot Motion and Control 2009*, Springer, Berlin/Heidelberg, pp. 297–306.
- Zefran, M. and Kumar, V. (1997). A variational calculus framework for motion planning, *IEEE International Conference on Robotics and Automation ICRA, Albuquerque, NM, USA*, pp. 415–420.
- Zhang, J. and Knoll, A. (1995). An enhanced optimization approach for generating smooth robot trajectories in the presence of obstacles, *Proceedings of the European Chinese Automation Conference, London, UK*, pp. 263–268.
- Zieliński, C. and Winiarski, T. (2010). Motion generation in the MRROC++ robot programming framework, *International Journal of Robotics Research* **29**(4): 386–413.



**Wojciech Szykiewicz** received the M.Sc. and Ph.D. degrees in automatic control and robotics from the Warsaw University of Technology, Poland. His current research activities concentrate on robot motion planning, manipulation and grasping, autonomous mobile robots, multi-robot/multi-agent systems, robot controller structures, and real-time and distributed systems. He works on sensor-based motion planning and control algorithms for multi-robot systems, including service, personal and mobile robots. He is the author or co-author of more than 70 publications.



**Jacek Błaszczyk** received his M.Sc. and Ph.D. degrees in automatic control from the Warsaw University of Technology, Poland, in 2000 and 2008, respectively. Currently, he is an assistant professor at the Research and Academic Computer Network (NASK), Poland. His research interest include large-scale nonlinear optimization, optimal control, parallel and distributed computations, numerical software for optimization and linear algebra, and, recently, modeling, simulation and optimization of large-scale water distribution systems.

Received: 26 October 2010

Revised: 29 May 2011