Nonlinear Optimization for Human-Like Movements of a High Degree of Freedom Robotics Arm-Hand System

Eliana Costa e Silva[1], Fernanda Costa[2], Estela Bicho[1], and Wolfram Erlhagen[2]

[1] Dept. of Industrial Electronics, University of Minho, Portugal
{esilva,estela.bicho}@dei.uminho.pt
http://www.dei.uminho.pt/
[2] Dept. of Mathematics and Applications, University of Minho, Portugal
{mfc,wolfram.erlhagen}@math.uminho.pt
http://www.math.uminho.pt/

Abstract. The design of autonomous robots, able to closely cooperate with human users in shared tasks, provides many new challenges for robotics research. Compared to industrial applications, robots working in human environments will need to have human-like abilities in their cognitive and motor behaviors. Here we present a model for generating trajectories of a high degree of freedom robotics arm-hand system that reflects optimality principles of human motor control. The process of finding a human-like trajectory among all possible solutions is formalized as a large-scale nonlinear optimization problem. We compare numerically three existing solvers, IPOPT, KNITRO and SNOPT, in terms of their real-time performance in different reach-to-grasp problems that are part of a human-robot interaction task. The results show that the SQP methods obtain better results than the IP methods. SNOPT finds optimal solutions for all tested problems in competitive computational times, thus being the one that best serves our purpose.

Keywords: anthropomorphic robotic system, reach-to-grasp, human-like collision-free arm movements, large-scale nonlinear optimization, interior-point methods, sequential quadratic programming.

1 Introduction

Robot motion planning problems have been studied for decades (for a survey on motion planning see [1,2]). However, most research concerns industrial robots in static and physically structured environments. In recent years, with the advances of information technology and mechanical design, there has been a remarkable change in the research focus of robotics applications, reaching beyond highly repetitive and high precision position tasks in industry. Currently, a new generation of service robots has started moving out of manufacturing environments into working environments such as homes, offices and hospitals that are shared with humans [3,4]. Prototype robotics systems have been tested for instance in

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 327–342, 2011.

elderly care, child education, rescue scenarios or assistance in our daily routines (for an overview of case studies see [5]). As fundamentally social beings, humans are experts in cooperating with others to achieve common goals in shared tasks [6]. In order to be accepted by a human user as a social partner an assistive robot has thus to show perceptual, cognitive and motor capacities that meet the expectancies the user might have about a pleasant and natural interaction. For instance, if a robot is supposed to work with the same objects and tools as its human partner, it may be beneficial for the team that the robot's arm and hand movements reflect invariant characteristics of human reaching and grasping trajectories. It has been argued that human-like movements greatly facilitate the interaction with robots since they allow the user to easily interpret the robot's movements in terms of goals [4,7,8]. The movements of robots in typical industrial applications are often perceived by humans as jerky and unrealistic. They are optimized to satisfy the specific needs of pre-defined tasks with essentially no interaction between human and robot.

In this paper we evaluate the real-time performance of a model for generating trajectories of a high degree of freedom (DOF) robotics arm-hand system that reflects optimality principles of human motor control and key characteristics of human arm trajectories [9]. The model has been implemented on the anthropomorphic robot **ARoS** (Fig. 1) and tested in different human-robot interaction tasks [10,11,12].

Fig. 1. The anthropomorphic robot **ARoS** engaged in human-robot collaboration

ARoS consists of a static torso, equipped with a 7 DOFs arm (shoulder - 3 DOF, elbow - 1 DOF, wrist - 3 DOF), a 4 DOFs three-fingered hand and a stereo vision system mounted on a pan-tilt unit [13]. Due to the redundant DOF of the arm and hand, a goal in everyday tasks like reach-to-grasp an object may be achieved in multiple ways. The selection of an optimal solution among all possible solutions, based on constraints that explain key characteristics of human arm trajectories in such tasks, gives rise to a large-scale nonlinear constrained optimization problem, since the time-continuous model is approximated by a finite dimensional problem obtained by discretizing the time. Very important for fluent and efficient human-robot interaction, the solution has to be found in real-time. Solving this type of large-scale problem has been recognized as a

big challenge in Optimization research as can be seen by the increasing number of academic and commercial solvers being developed in recent years (see for example [14,15,16]). These solvers implement different constrained optimization algorithms such as Sequential Quadratic Programming (SQP) methods and Interior Point (IP) methods, among others. We have selected three solvers that are adequate for the purpose of this study, namely, IPOPT [14], KNITRO [15] and SNOPT [16], since they are all well regarded and recognized solvers for large-scale nonlinear optimization. Additionally they all accept as input an optimization problem written in the AMPL[1] modelling language. This language provides an interface that allows the user to easily choose among solvers and options that may improve solver performance. Perhaps the major point in favor of the use of AMPL is that it provides a common mechanism for convening problem codes to solve them and the user does not need to specify derivatives of the objective and constraints functions. Here we compare the performance of the three solvers in terms of their capacity to find in real-time collision-free arm and hand trajectories in various object grasping tasks.

The paper is organized as follows. In Sect. 2 we give a brief overview about the three nonlinear constrained optimization solvers and highlight their differences. The formalization of the movement planning problem as a nonlinear constrained optimization problem is presented in Sect. 3. In Sect. 4 we systematically compare the numerical results provided by the different solvers in four reach-to-grasp problems that are part of a human-robot interaction task. The paper finishes with a discussion of conclusions and future work.

2 Nonlinear Constrained Optimization Solvers

We have compared the performance of the following three solvers: IPOPT, implements an IP filter line search method [14]; KNITRO, implements both an interior-point and an active-set sequential linear-quadratic programming (SLQP) trust region methods [15]; SNOPT, implements an active-set SQP line search method [16]. The three general nonlinear constrained solvers assume that the nonlinear objective and constraints functions are smooth and that their first derivatives are available. IPOPT and KNITRO in addition assume the availability of the second derivatives.

IPOPT is an open source software package for large scale nonlinear optimization, that implements a primal-dual barrier method for solving nonlinear optimization problems. The optimal solution is obtained by computing approximate solutions of a sequence of (associated) barrier problems for a decreasing sequence of barrier parameters converging to zero. To promote global convergence, IPOPT employs a line-search filter strategy when solving each barrier problem [14].

KNITRO is an optimization software library, that implements both an interior-point method and an active-set method for solving the nonlinear optimization

[1] `http://www.ampl.com/`

problems [15]. In the interior method, the nonlinear problem is replaced by a sequence of barrier problems controlled by a barrier parameter, $\mu \to 0$, and is similar in the concept to IPOPT. The algorithm uses trust region and a merit function to promote global convergence. KNITRO also implements an active-set SLQP algorithm, and is similar in nature to a SQP method but uses linear programming sub-problems to estimate the active-set at each iteration.

SNOPT is a SQP algorithm that uses an active-set approach for solving large nonlinearly constrained optimization problems. The central feature of a SQP method is that the search directions are the solutions of quadratic programming subproblems that minimize a quadratic model of the Lagrangian function subject to linearized constraints. SNOPT is a first-order code, employing a limited-memory quasi-Newton approximation for the Hessian of the Lagrangian, namely, the BFGS method. To guarantee convergence from any starting point, SNOPT uses a line search with an augmented Lagrangian merit function [16].

3 Movement Planning as a Nonlinear Constrained Optimization Problem

In this section we formalize the movement planning of the anthropomorphic robot **ARoS** as a discrete time model which results in a large-scale nonlinear optimization problem with simple bounds and equality and inequality constraints.

To develop this movement planning model we got inspiration from observed regularities in human upper-limb movement studies and on models proposed by the human motor control community, specially the posture-based motion planning model by Rosenbaum and colleagues [17]. This model proposes that when planning reaching and grasping movements, humans subdivide this problem into two subproblems: final posture selection and trajectory selection. Additionally, these authors propose that obstacle avoidance is achieved by the superimposition of two movements: a direct movement from the initial posture to the final posture, and a bounce movement from the initial to a bounce posture and back. This bounce posture serves as a subgoal for a *back-and-forth* movement, which is superimposed on the direct movement from initial to final posture.

A robotic arm (and hand) can be represented as a series of links connected by joints. The number of joints which can be independently actuated define its DOFs. Each DOF is associated to an independent variable, θ_k, where k is the number of DOF. **ARoS**' anthropomorphic robotic arm has 7 DOFs and its hand has 4 DOFs. The arm and hand configuration in joint space is thus completely defined by the vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{11})^\top$.

3.1 Problem Formulation

The movement planning proposed here for the anthropomorphic robot can be summarised as the resolution of two subproblems. First we need to find the final posture, i.e., a vector of arm and hand joint angles, $\boldsymbol{\theta}_f \in \mathbb{R}^{11}$, that allows **ARoS** to grasp a given object subject to specific constraints (e.g. grip type). A direct

movement from initial to final posture, with a bell-shaped unimodal velocity profile is then generated. The second subproblem consists of determining a bounce posture, $\boldsymbol{\theta}_b \in \mathbb{R}^{11}$, that serves as a sub-goal for a *back-and-forth* movement, to be superimposed on the direct movement, for avoiding collision with obstacles in the robot's workspace. The sequence of joint angles of the robotics arm and hand is given by

$$\begin{aligned}\boldsymbol{\theta}(t,\boldsymbol{\theta}_f,\boldsymbol{\theta}_b) = \boldsymbol{\theta}_0 + (\boldsymbol{\theta}_f - \boldsymbol{\theta}_0)\left(10\tau^3 - 15\tau^4 + 6\tau^5\right) + \boldsymbol{v}_0\, T\left(\tau - 6\tau^3 + 8\tau^4 - 3\tau^5\right)\\ + \frac{1}{2}\boldsymbol{a}_0\, T^2\left(\tau^2 - 3\tau^3 + 3\tau^4 - \tau^5\right) + (\boldsymbol{\theta}_b - \boldsymbol{\theta}_0)\sin^2(\pi\,\tau^{\vartheta}),\end{aligned} \tag{1}$$

where $\boldsymbol{\theta}_0, \boldsymbol{v}_0, \boldsymbol{a}_0 \in \mathbb{IR}^{11}$ are constant vectors representing initial joint position, velocity and acceleration, respectively, $T \in \mathbb{IR}^{+}$ represents the movement duration, $t \in [0,T]$, $\tau = \frac{t}{T} \in [0,1]$ is the normalized movement duration, and $\vartheta = -\frac{\ln 2}{\ln t_b}$, $t_b \in]0,1[$ is the movement time when the bounce posture is applied. This trajectory parametrization will be used in the definition of the obstacle avoidance constraints (c.f. (7) and (14) in Sect. 3.4). The parameters $\boldsymbol{\theta}_f \in \mathbb{IR}^{11}$ and $\boldsymbol{\theta}_b \in \mathbb{IR}^{11}$, i.e., the final and the bounce posture of the robotics arm and hand, are the solution of two nonlinear constrained optimization problems.

In a first step we determine $\boldsymbol{\theta}_f \in \mathbb{IR}^{11}$ as the posture that (i) allows the object to be successfully grasped with the grip type that satisfies the action intention[2], and (ii) minimizes the displacements of the joints from the beginning to the end of the movement,

$$\text{(Pa)} \quad \min_{\boldsymbol{\theta}_f \in \Theta_f \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{f,k}\right)^2, \lambda_k \geq 0,$$

where $\Theta_f \subset \mathbb{R}^{11}$ is the set of all admissible postures that permit the object to be successfully grasped with the desired grip type. Next, and using $\boldsymbol{\theta}_f$ determined previously, we determine $\boldsymbol{\theta}_b \in \mathbb{IR}^{11}$ as the posture that (i) yields a collision-free movement from start to end and (ii) presents minimum displacement of the joints,

$$\text{(Pb)} \quad \min_{\theta_b \in \Theta_b \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{b,k}\right)^2, \lambda_k \geq 0,$$

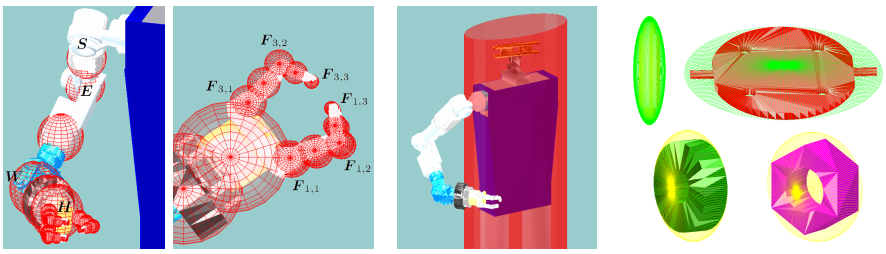
where Θ_b is the set of all admissible bounce postures of the arm and hand that yields collision-free movements. We discretize $t \in [0,T]$ by N_T equally spaced points $t_i = i\,h$, where $h = \frac{T}{N_T}$ is the step size and $i = 0, 1, \ldots, N_T$. Our convention is that $\boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ represents $\boldsymbol{\theta}(t, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ at t_i. Before proceeding to the effective specification of the constraints that define Θ_f and Θ_b, in problems (Pa) and (Pb), respectively, we present the modelling of the robot's body (i.e., arm, hand and torso) and of the objects in its workspace.

[2] The grip type, i.e., how the object should be grasped, is selected by the Cognitive Model of the robot **ARoS** by taking into account the action intention [11,12].

3.2 Modelling of Robot and Objects

Robot. ARoS' robotics arm and hand are composed of a series of links connected in pairs by rotational joints. To each joint i is attached a local frame $\hat{\boldsymbol{x}}_i\hat{\boldsymbol{y}}_i\hat{\boldsymbol{z}}_i$. For translations and rotations description of the robotic arm and hand we use the Denavit-Hartenberg parameters [18].

For describing the nonlinear inequality constraints we have considered the arm and hand composed of 21 spheres, as shown in Fig. 2(a), with radius given by the arm and hand dimensions and whose centers are written as functions of the joint angles.



(a) (b) (c)

Fig. 2. **ARoS**' robotics arm and hand are modelled as 21 spheres(a); its torso is modelled as an elliptic cylinder(b); the objects (see Fig. 1) are modelled as ellipsoids(c)

We start by determining principal points on the robotics arm and hand, namely, shoulder, $\boldsymbol{S}$, elbow, $\boldsymbol{E}$, wrist, $\boldsymbol{W}$, palm of the hand, $\boldsymbol{H}$, and for each robotic hand finger $k = 1, 2, 3$, the points $\boldsymbol{F}_{k,1}, \boldsymbol{F}_{k,2}$ and $\boldsymbol{F}_{k,3}$. The 3D position of these points are nonlinear functions of joint angles given by forward kinematics. Hand orientation is defined using the local frame $\hat{\boldsymbol{x}}_7\hat{\boldsymbol{y}}_7\hat{\boldsymbol{z}}_7$ at the arm's last joint, where $\hat{\boldsymbol{x}}_7, \hat{\boldsymbol{y}}_7, \hat{\boldsymbol{z}}_7 : \mathbb{R}^7 \longrightarrow \mathbb{R}^3$ are nonlinear functions of arm's joint angles.

We model **ARoS**' torso as an elliptic cylinder, i.e., $\left(\frac{x-x_0}{a}\right)^2 + \left(\frac{y-y_0}{b}\right)^2 \leq 1$, with $x_0 = 0$, $y_0 = 50$, $a = 170$ and $b = 350$ mm (see Fig. 2(b)).

Objects in the Robot's Workspace. Each object in the robot's workspace is defined by its center position, $\boldsymbol{C} = (x_c, y_c, z_c)^\top \in \mathbb{R}^3$, and its orientation, relative to a external world frame, described by the Euler angles[3] ϕ, ψ and γ. Using the orientation and position of the object we define a local frame $\hat{\boldsymbol{x}}\hat{\boldsymbol{y}}\hat{\boldsymbol{z}}$ attached to it. The orientation may also be defined by the rotation matrix $\boldsymbol{R} = \boldsymbol{R}(\phi, \psi, \gamma) = [\hat{\boldsymbol{x}} \mid \hat{\boldsymbol{y}} \mid \hat{\boldsymbol{z}}]$. In addition to the position and orientation of the object, we have its dimensions on the main three axis R_x, R_y and R_z.

For imposing the constraints for avoiding obstacles (see Sect. 3.3) we have considered two different models: (i) the object is modelled as a set of spheres in its interior with radius determined by $R_s = \min(R_x, R_y, R_z)$; (ii) the object is modelled as an ellipsoid enclosing it, defined as the quadratic inequality,

[3] Also called roll, pitch, yaw fixed-axis rotations.

$(\boldsymbol{X}-\boldsymbol{C})^\top \boldsymbol{R}^\top \boldsymbol{A}\boldsymbol{R}(\boldsymbol{X}-\boldsymbol{C}) \leq 1$, where $\boldsymbol{A} = \text{diag}((R_x)^{-2}, (R_y)^{-2}, (R_z)^{-2})$. Some examples of ellipsoids enclosing objects are depicted in Fig. 2(c).

3.3 Specifying Obstacles Constraints

For simplicity in this section we use the notation $\boldsymbol{\theta} \equiv \boldsymbol{\theta}_f$ and $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ to refer to the inequality constraints (7) and (14), respectively. Let $\boldsymbol{P}_k(\boldsymbol{\theta}) = (P_{k,1}(\boldsymbol{\theta}), P_{k,2}(\boldsymbol{\theta}), P_{k,3}(\boldsymbol{\theta}))^\top, k = 1, \ldots, 21$, be the centers of the 21 spheres on the robotics arm and hand as described in Sect. 3.2. Additionally, let $n_{obj} \in \mathbb{N}$ be the number of objects (obstacles and target) in the robot's workspace, $\boldsymbol{C}_l, l = 1, \ldots, n_{obj}$, be their centers, $n_{sph} \geq n_{obj}$ be the total number of spheres in the interior of these objects and $\boldsymbol{O}_j, j \in 1, \ldots, n_{sph}$, their centers.

The inequality constraints (7) and (14) are due to obstacle avoidance, namely, collision between: (constr1) body and arm/hand; (constr2) table and arm/hand; (constr3) obstacles and arm/hand; and (constr4) target object and arm/hand. Constraints for avoiding body and arm/hand superposition, (constr1), are

$$\left(\frac{P_{k,1}(\boldsymbol{\theta}) - x_0}{a}\right)^2 + \left(\frac{P_{k,2}(\boldsymbol{\theta}) - y_0}{b}\right)^2 - 1 \geq 0, k = 1, \ldots, 21.$$

Constraints (constr2) are given by

$$P_{k,3}(\boldsymbol{\theta}) - r_k - h_{table} \geq 0, k = 1, \ldots, 21,$$

where h_{table} is the table's height and r_k is the radius of the sphere with center at $\boldsymbol{P}_k(\boldsymbol{\theta})$. Constraints (constr3) and (constr4) can be defined by modelling the objects as a set of spheres or ellipsoids. For spheres, the constraints are

$$\| \boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{O}_j \|_2^2 - (r_k + R_j + \varepsilon)^2 \geq 0, k = 1, \ldots, 21, j = 1, \ldots, n_{sph}, \qquad (2)$$

where r_k and R_j are the arm/hand and obstacles radius, respectively, and $\varepsilon > 0$ is a clearance tolerance. If each object $l = 1, \ldots, n_{obj}$, is modelled as a single ellipsoid, then constraints (constr3) and (constr4) take the form

$$(\boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{C}_l)^\top \boldsymbol{R_l}^\top \boldsymbol{A}_l \boldsymbol{R_l} (\boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{C}_l) - 1 \geq 0, \;\; k = 1, \ldots, 21, \qquad (3)$$

where $\boldsymbol{A_l} = \text{diag}((r_k + R_{x,l} + \varepsilon)^{-2}, (r_k + R_{y,l} + \varepsilon)^{-2}, (r_k + R_{z,l} + \varepsilon)^{-2})$, and $R_{x,l}$, $R_{y,l}$, $R_{z,l}$ are the object's dimensions in its main three axis.

Defining the obstacle constraints using (2), leads to a very large number of constraints and some solvers fail due to insufficient memory allocation. Therefore, the obstacles constraints for the problems in Sect. 4 were defined by (3).

3.4 Specifying the Problems

The nonlinear constrained optimization problem (Pa) is defined at a single instant in time, that is, the instant when the robot's hand is in contact with the object to be grasped, and is generally written as:

$$\text{(Pa)} \min_{\boldsymbol{\theta}_f \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{f,k}\right)^2, \lambda_k \geq 0 \tag{4}$$

subject to

$$\boldsymbol{H}(\boldsymbol{\theta}_f) + d_{HO}(\theta_{f,9})\ \hat{\boldsymbol{z}}_7(\boldsymbol{\theta}_f) - \boldsymbol{X}_{tar} = \boldsymbol{0} \tag{5}$$

$$\hat{\boldsymbol{x}}_7(\boldsymbol{\theta}_f) - \hat{\boldsymbol{z}}_{tar} = \boldsymbol{0} \tag{6}$$

$$\boldsymbol{h}_f(\boldsymbol{\theta}_f) \geq \boldsymbol{0} \tag{7}$$

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}_f \leq \boldsymbol{\theta}_{max} \tag{8}$$

$$\theta_{f,8} = 0, \theta_{f,9} = \theta_{f,10} = \theta_{f,11} \tag{9}$$

where $\boldsymbol{\theta}_{min}, \boldsymbol{\theta}_{max}$ are constant vectors that represent the lower and upper mechanical joint limits, $\boldsymbol{X}_{tar} \in \mathbb{R}^3$ is the position and $\hat{\boldsymbol{x}}_{tar}\hat{\boldsymbol{y}}_{tar}\hat{\boldsymbol{z}}_{tar}$ is the local frame attached to the object that the robot must grasp, $d_{HO}(\theta_{f,9})$ is the distance from the object's center to the palm of the robotics hand. The equality constraints (6) depend on the desired grip type. The nonlinear inequality constraints defined by (7) are due to obstacle avoidance and were formalized in Sect. 3.3. The joint angles of the fingers (9) are determined by the spatial dimensions of the object and the grip type.

The nonlinear optimization problem (Pb) is stated as:

$$\text{(Pb)} \min_{\boldsymbol{\theta}_b \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{b,k}\right)^2, \lambda_k \geq 0 \tag{10}$$

subject to

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) \leq \boldsymbol{\theta}_{max} \tag{11}$$

$$\theta_8(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) = 0 \tag{12}$$

$$\theta_9(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) = \theta_{10}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) \tag{13}$$

$$\boldsymbol{h}_b(\boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)) \geq \boldsymbol{0}, \qquad t_i = 0, \ldots, T \tag{14}$$

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}_b \leq \boldsymbol{\theta}_{max}\ . \tag{15}$$

4 Results

The performance of the three nonlinear optimization solvers has been tested in four different problems (see Table 1), in which the anthropomorphic robot **ARoS** has to grasp different objects, with different grip types, thereby avoiding several obstacles. Specifically, we focus here on reaching and grasping columns and wheels, using a side and an above grip, respectively (see Fig. 1). The grasping behaviors are part of a joint assembly task described in detail in [10,12,11]. In each of these problems, $\# = 1, 2, 3, 4$, we need to solve two nonlinear optimization subproblems:

- (P#a) the selection of the final posture as defined by (4)-(9);
- (P#b) the selection of the bounce posture as defined by (10)-(15).

The numerical results were obtained using an Intel(R) Core(TM)2 Duo CPU P7450@ 2.13GHz running Windows 7 64 bits. All problems are coded in

Table 1. Problems description

Problem	Object to be grasped	Other objects in the robot's workspace	Grip type	Equality constraint (6)
1	Column	Table and base	Side	$\hat{\boldsymbol{x}}_7(\boldsymbol{\theta}_f) - \hat{\boldsymbol{z}}_{tar} = \mathbf{0}$
2		Table, base and column		
3	Wheel	Table and base	Above	$\hat{\boldsymbol{z}}_7(\boldsymbol{\theta}_f) + \hat{\boldsymbol{z}}_{tar} = \mathbf{0}$
4		Table, base and column		

AMPL and solved using KNITRO 7.0.0, SNOPT 7.2-8 and IPOPT 3.8.0. We use the default options that can be found at the URLs: IPOPT - `http://projects.coin-or.org/Ipopt`; KNITRO - `http://www.ziena.com/`; SNOPT - `http://www.sbsi-sol-optimize.com/`. The default termination tolerance for KNITRO and KNOPT is 10^{-6}, thus we set IPOPT's termination tolerance to the same value. We present the results for both the IP and SQP version of KNITRO. Since SNOPT uses a limited-memory BFGS, we also tested IPOPT and KNITRO with this option. The movement of the robot's arm and hand is shown by a software simulator written in Matlab, that uses a CAD model of the real robotic platform, including its torso, robotic arm and hand and camera head.

For large scale problems the main computational burden is to solve a linear system at each iteration that is required to compute the regular step. The IPOPT 3.8.0 was run with the linear solver MUMPS. We also tested the linear solver HSL MA27, but IPOPT's performance was similar. The IP version of KNITRO was run with the linear solver HSL MA57, while the SQP version uses HSL MA27. SNOPT uses a Cholesky factorization to solve the linear system.

In Tables 2-5 we present the CPU time (in sec.), the number of iterations and the value of the objective function after AMPL presolve. Further, we report the number of variables, N, the number of equality, M_{eq}, and inequality constraints, M_{ineq}, the percentage of nonzero elements in the Jacobian and Hessian matrices. KNITRO-IP and KNITRO-SQP stands for IP and SQP versions of KNITRO, respectively, "(exact)" means that solvers use exact second order derivatives information, while "(L-BFGS)" means that these are approximated using a limited-memory Broyden-Fletcher-Goldfarb-Shanno method. In our implementations the value of the following parameters were fixed: $N_T = 30$, $T = 1$, $t_b = 0.5$ and $\lambda_k = 1, k = 1, \ldots, 11$.

4.1 Problem 1

In the first problem, **ARoS** has to grasp a column that is hold out for the robot by the human partner (see right panel in Fig. 1). The numerical results are reported in Table 2.

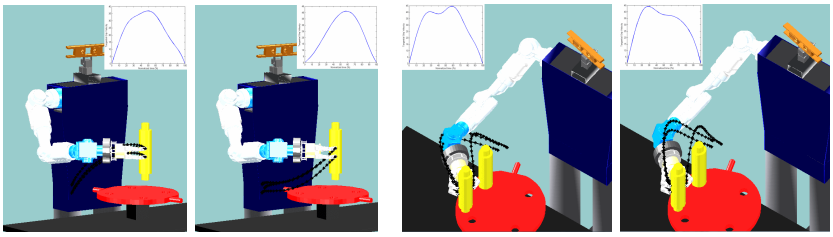
Table 2. Numerical results for Problem 1

	Number of iterations	Objective	CPU (sec.)
(P1a) ($N = 7, M_{eq} = 6, M_{ineq} = 41$; Nonzero elements: Jacob 86%, Hess 57%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-IP (exact)	140	4.649723765	0.203
KNITRO-IP (L-BFGS)	97 *i)*	4.649722112	0.140
KNITRO-SQP (exact)	48	4.649723765	0.218
KNITRO-SQP (L-BFGS)	12	4.649723765	0.047
SNOPT	21	4.649723766	<0.01
(P1b) ($N = 10, M_{eq} = 0, M_{ineq} = 1819$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	138	0.894267104	12.541
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	32 *i)*	0.913400026	7.379
KNITRO-SQP (L-BFGS)	47	0.8942674282	2.075
SNOPT	145	0.7954218878	0.42

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

Subproblem (P1a). This is a dense small-scale problem (see Table 2). For the determination of the final posture, SNOPT, KNITRO-SQP and KNITRO-IP find an optimal solution. With KNITRO-IP (L-BFGS) the stop criteria is not reached. However, it converges to a solution (i.e., a final posture) that is very close to the optimal solution found by the other solvers. In terms of robot overt behavior, the difference between the solutions can be neglected since the resolution of the robot's joints is inferior to 0.05 rad. SNOPT is the fastest with a CPU time inferior to 0.01 sec.

Subproblem (P1b). The best solution for problem (P1a), i.e., the selected optimal final posture, is used in problem (P1b), i.e., for determining the bounce posture. This is a dense large-scale problem (see Table 2). The numerical results reported in Table 2 show that IPOPT (L-BFGS), SNOPT and KNITRO-SQP (L-BFGS) are able to find an optimal solution. The best solution is found by SNOPT. Although IPOPT (L-BFGS) and SNOPT take nearly the same number of iterations, in terms of CPU time SNOPT performs better. The KNITRO-IP and IPOPT (exact) exceed the maximum number of iterations. Although KNITRO-SQP (exact) does not satisfy the termination criteria, it converges to a feasible point (as can be verified by executing the resultant joint trajectory using the simulator). The movement of the robotics arm and hand for the two optimal solutions are depicted in Fig. 3.



(a) Problem 1 (b) Problem 2

Fig. 3. ARoS has to reach and grasp a column from the hand of its human partner (not shown). Panel (a) shows the two different trajectories found for Problem 1. The solution found by KNITRO-SQP (L-BFGS) and IPOPT (L-BFGS) (left) is significantly different than the one found by SNOPT (right). Panel (b) shows the trajectories for Problem 2 obtained by IPOPT (L-BFGS) (left) and by SNOPT (right). The trajectories are smooth and collision-free. They display characteristics found in human upper limb movements such as for e.g. almost bellshaped continuous hand velocity and a second velocity peak for obstacle avoidance.

4.2 Problem 2

This problem is similar to the first one, however there is an additional obstacle (i.e. a column attached to the base) that makes the movement found before infeasible since it would result in collision with the new object.

Subproblem (P2a). This is a dense small-scale problem (see Table 3). For this problem the SQP methods present a higher performance than the IP methods that did not find an optimal solution. In fact, SNOPT and KNITRO-SQP converge to the same optimal point and once again SNOPT is the fastest.

Subproblem (P2b). For the determination of the bounce posture), IPOPT (exact) and the KNITRO-IP exceed the number of iterations. SNOPT, once again found an optimal solution in the smallest CPU time. This is a dense large-scale problem (see Table 3). The KNITRO-SQP reaches a nearly optimal point. Figure 3 depicts the movement of the robotics arm and hand for the two optimal solutions found.

4.3 Problem 3

In the third test problem, **ARoS** must reach and grasp a wheel that is hold out by the human partner (see left panel in Fig. 1). The grasping behavior is now different from the previous problems (see Table 1).

Subproblem (P3a). This is a dense small-scale problem (see Table 4). All the solvers, with the exception of KNITRO-IP (L-BFGS), find the same optimal solution in less than approximately 0.5 secs. The best results in terms of CPU

Table 3. Numerical results for Problem 2

	Number of iterations	Objective	CPU (sec.)
(P2a) ($N = 7, M_{eq} = 6, M_{ineq} = 62$; Nonzero elements: Jacob 89%, Hess 57%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-IP (exact)	657[i)]	4.649723765	1.186
KNITRO-IP (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-SQP (exact)	39	4.649723765	0.203
KNITRO-SQP (L-BFGS)	9	4.649723765	0.031
SNOPT	500	4.649723763	0.06
(P2b) ($N = 10, M_{eq} = 0, M_{ineq} = 2429$; Nonzero elements: Jacob 97%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	450	1.013981558	60.242
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	10[i)]	1.498687348	3.682
KNITRO-SQP (L-BFGS)	50[i)]	1.007519005	35.397
SNOPT	115	1.031100189	0.57

[i)] Relative change in feasible solution estimate < xtol (KNITRO). [ii)] Maximum number of iterations reached. [iii)] Converged to a locally infeasible point.

Table 4. Numerical results for Problem 3

	Number of iterations	Objective	CPU (sec.)
(P3a) ($N = 7, M_{eq} = 6, M_{ineq} = 34$; Nonzero elements: Jacob 88%, Hess 57%)			
IPOPT (exact)	210	6.855991978	0.511
IPOPT (L-BFGS)	21	6.855991887	0.044
KNITRO-IP (exact)	14	6.855991912	0.016
KNITRO-IP (L-BFGS)	114	23.22895729	0.125
KNITRO-SQP (exact)	6	6.855991907	0.031
KNITRO-SQP (L-BFGS)	10	6.855991907	0.031
SNOPT	495	6.855991907	0.34
(P3b) ($N = 10, M_{eq} = 0, M_{ineq} = 1819$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	138	0.743876763	11.294
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	13	0.7438768249	1.357
KNITRO-SQP (L-BFGS)	440[i)]	0.745302671	16.115
SNOPT	85	0.7364848552	0.30

[i)] Relative change in feasible solution estimate < xtol (KNITRO). [ii)] Maximum number of iterations reached. [iii)] Converged to a locally infeasible point.

time is found by KNITRO-IP (exact). SNOPT presents the highest number of iterations and the second longer CPU time. The local optimum found by KNITRO-IP (L-BFGS) (Figure 4(a), left panel) is more costly and represents an ackward posture. On the contrary, the optimal solution found by the other solvers is quite pleasant (Figure 4(a), right panel).

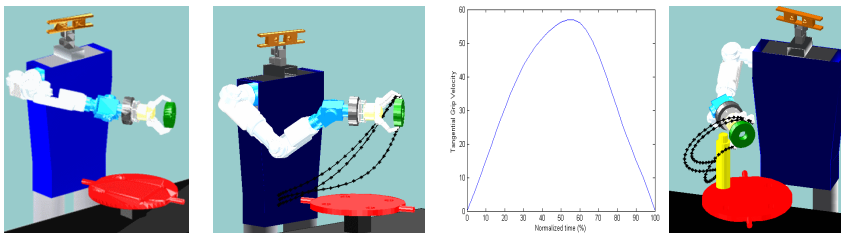
Subproblem (P3b). This is a dense large-scale problem (see Table 4). For computing the bounce posture, the best performance is obtained by SNOPT, both in terms of the objective value and CPU time. IPOPT (L-BFGS) and KNITRO-SQP (exact) converge to approximately the same optimal solution. Figure 4(a) depicts the movement of the robotics arm and hand movement that represents the best solution.

4.4 Problem 4

In the last test problem **ARoS** has to reach and grasp a wheel that is hold out by the human, while avoiding collision with a column attached to the base.

Subproblem (P4a). This is a dense small-scale problem (see Table 5). For the selection of the final posture, IPOPT (L-BFGS), KNITRO-IP (exact), KNITRO-SQP and SNOPT find the same optimal solution. KNITRO-IP (L-BFGS) exceed the maximum number of iterations and IPOPT (exact) converges to an infeasibility point. KNITRO-IP (exact) is the fastest.

Subproblem (P4b). This is a dense large-scale problem (see Table 5). For the bounce posture selection, only SNOPT (see Fig. 4 for the movement of the robotic arm and hand) finds an optimal solution in 94 iterations and 0.46 secs. The solutions obtained by KNITRO-SQP are nearly optimal.



(a) Problem 3 (b) Problem 4

Fig. 4. ARoS has to reach and grasp a wheel from the hand of its human partner (not shown). The two snapshots in panel (a) show the final postures of the best solution (right) and the solution with the highest cost (left), found for Problem 3. Panel (b) shows the tangential hand velocity (left) and the trajectory (right), found for Problem 4.

Table 5. Numerical results for Problem 4

	Number of iterations	Objective	CPU (sec.)
(P4a) ($N = 7, M_{eq} = 6, M_{ineq} = 52$; Nonzero elements: Jacob 88%, Hess 57%)			
IPOPT (exact)	*iii)*	*iii)*	*iii)*
MUMPS IPOPT (L-BFGS)	130	6.855991887	0.389
MUMPS KNITRO-IP (exact)	13	6.855991912	0.016
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	6	6.855991907	0.031
KNITRO-SQP (L-BFGS)	10	6.855991907	0.031
SNOPT	494	6.855991907	0.52
(P4b) ($N = 10, M_{eq} = 0, M_{ineq} = 2428$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	307*i)*	0.805153979	42.853
KNITRO-SQP (L-BFGS)	45*i)*	0.807075699	3.822
SNOPT	94	0.7982104054	0.46

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

5 Discussion and Future Work

In this paper we have presented a model for planning trajectories of a high DOFs robotics arm-hand system in reach-to-grasp tasks. Since a main motivation for this work is to guarantee human-like motion, the model takes into account regularities and optimality principles of human upper-limb movements observed in behavioral studies. The robot's overt behavior exhibits key characteristics of human movement, such as, smooth, fluent and graceful movements, slightly curved path of the hand, maximal finger aperture occuring during the second half of the movement, biphasic tangential velocity profile [19]. The formalization of the model as a two-step optimization problem in posture space is inspired by Rosenbaum's planning model that has been applied in the past to qualitatively explain human reach-to-grasp trajectories [17]. The problem of generating realistic trajectories is formalized as a general nonlinear and nonconvex optimization problem with simple bounds, equality and inequality constraints.

Four real-world problems have been taken from a recent human-robot interaction (HRI) study in which the anthropomorphic robot **ARoS** assembles toy objects together with a human partner [10,11,12]. For each a dense small-scale subproblem (determining the final posture) and a dense large-scale subproblem (determining the bounce posture) have to be solved. These optimization problems must be solved in real-time in order to guarantee fluent HRIs.

We have compared the performance of three state-of-the-art solvers, IPOPT, KNITRO and SNOPT, that have proven in the past their computational power in

several large-scale test problems. A major difference between the solvers concerns the optimization techniques used, namely, IP and SQP methods. In general, for the problems tested in this paper, the SQP methods seem to outperform the IP methods. SNOPT was able to find an optimal solution for all test problems which was not the case for IPOPT and KNITRO. In addition, SNOPT took less CPU time compared to the other solvers in most cases.

The good results of SNOPT obtained in a trajectory generation problem with many constraints may be explained by the fact that SNOPT works on a reduced space of the variables by using the constraints. Another difference between the solvers that may contribute to the superior performance of SNOPT is that contrary to IPOPT (exact) and KNITRO (exact), that require the second order derivatives, SNOPT only needs first order derivatives of the objective and constraints functions. In IPOPT and KNITRO we can also approximate the second-order derivatives by a limited-memory BFGS. For the problems addressed in this paper using the limited-memory BFGS quasi-Newton method has proven to be more competitive than a Newton approach.

The simulation studies show that SNOPT found optimal trajectories in all tested problems in less than 1 sec. This makes SNOPT a good candidate for a NL solver that guarantees real-time performance in real-world HRI tasks. It is important to stress that a fine tuning of parameters of the solvers in order to reduce CPU time was beyond the scope of this comparison study. It is clear that for instance the selection of a more sophisticated stopping criteria may further improve the real-time solvers performance. We plan to test SNOPT as part of the control architecture of **ARoS** in the nearer future in different HRI tasks.

A limitation of the tested solvers is that they only guarantee convergence to a local optimum. It would be highly interesting to use global optimization software in the future.

We believe that the complex problem of trajectory generation in real-time, real-word robotics applications provide a rich and fertile ground for new research in nonlinear optimization.

Acknowledgement

Eliana Costa e Silva was supported by FCT (grant: SFRH/BD/23821/2005). The resources and equipment were financed by FCT and UM through project "Anthropomorphic robotic systems: control based on the processing principles of the human and other primates' motor system and potential applications in service robotics and biomedical engineering" (Ref. CONC-REEQ/17/2001) and by EC through project "JAST: Joint-Action Science and Technology" (Ref. IST-2-003747-IP). We thank the Mobile and Anthropomorphic Robotics Laboratory at University of Minho for constant good work environment. Finally, we would like to thank Carl Laird and Andreas Wächter for making available IPOPT, and AMPL for making available an unrestricted 30 days trial version of AMPL, KNITRO and SNOPT executables.

References

1. Hwang, Y.K., Ahuja, N.: Gross motion planning a survey. ACM Computing Surveys (CSUR) 24(3), 219–291 (1992)
2. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
3. Schraft, R.D., Schmierer, G.: Service Robots: products, scenarios, visions. A K Peters, Ltd., Wellesley (2000)
4. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots: concepts, design. Robot. Auton. Syst. 42(3-4), 143–166 (2003)
5. Kiesler, S., Hinds, P.: Human-Robot Interaction. Special Issue of Hum-Comput Interact 19(1/2) (2004)
6. Sebanz, N., Bekkering, H., Knoblich, G.: Joint action: bodies and minds moving together. Trends Cogn. Sci. 10(2), 70–76 (2006)
7. Duffy, B.R.: Anthropomorphism and the social robot. Robot Auton. Syst. 42(3-4), 177–190 (2003)
8. Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, S., Valavanis, K., Vukobratovic, M.: How far away is artificial man? IEEE Robot. Autom. Mag. 8(1), 66–73 (2001)
9. Todorov, E.: Optimality principles in sensorimotor control. Nat. Neurosci. 7(9), 907–915 (2004)
10. Bicho, E., Louro, L., Hipólito, N., Erlhagen, W.: A dynamic field approach to goal inference and error monitoring for human-robot interaction. In: Dautenhahn, E. (ed.) Proceedings of the 2009 Inter. Symp. on New Frontiers in HRI. AISB Convention, April 8-9, pp. 31–37. Heriot-Watt University, Edinburgh (2009)
11. Bicho, E., Erlhagen, W., Louro, L., Costa e Silva, E., Silva, R., Hipólito, N.: A dynamic field approach to goal inference, error detection and anticipatory action selection in human-robot collaboration. In: Sanders, J., Dautenhahn, K. (eds.) New Frontiers in Human-Robot Interaction. John Benjamins Publishing Company, Amsterdam (accepted)
12. Bicho, E., Erlhagen, W., Louro, L.: Costa e Silva, E.: Neuro-cognitive mechanisms of decision making in joint action: a Human-Robot interaction study. Hum. Movement. Sci (January 3, 2011) doi:10.1016/j.humov.2010.08.012
13. Silva, R., Bicho, E., Erlhagen, W.: **ARoS**: An Anthropomorphic Robot For Human-Robot Interaction And Coordination Studies. In: CONTROLO 2008, pp. 819–826 (2008)
14. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57 (2007)
15. Byrd, R., Nocedal, J., Waltz, R.: Knitro: An integrated package for nonlinear optimization. Large Scale Nonlinear Optimization, 35–59 (2006)
16. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM J. Optimiz. 12, 979–1006 (2002)
17. Rosenbaum, D., Meulenbroek, R., Vaugham, J., Jansen, C.: Posture-based Motion planning: Applications to grasping. Psychol. Rev. 108(4), 709–734 (2001)
18. Craig, J.J.: Introduction to robotics: mechanics and control, 2nd edn. Addison-Wesley, Reading (1998)
19. Lommertzen, J., Costa e Silva, E., Cuijpers, R.H., Meulenbroek, R.G.J.: Collision-avoidance characteristics of grasping. In: Pezzulo, G., Butz, M.V., Sigaud, O., Baldassarre, G. (eds.) ABiALS 2008. LNCS, vol. 5499, pp. 188–208. Springer, Heidelberg (2009)