

# Human-Like Movement of an Anthropomorphic Robot: Problem Revisited

E. Costa e Silva<sup>\*</sup>, M. F. Costa<sup>†</sup>, E. Bicho<sup>\*</sup> and W. Erhagen<sup>†</sup>

<sup>\*</sup>*Dept. of Industrial Electronics, University of Minho, Portugal*

<sup>†</sup>*Dept. of Mathematics and Applications, University of Minho, Portugal*

**Abstract.** Human-like movement is fundamental for natural human-robot interaction and collaboration. We have developed in a model for generating arm and hand movements an anthropomorphic robot. This model was inspired by the Posture-Based Motion-Planning Model of human reaching and grasping movements. In this paper we present some changes to the model we have proposed in [4] and test and compare different nonlinear constrained optimization techniques for solving the large-scale nonlinear constrained optimization problem that rises from the discretization of our time-continuous model. Furthermore, we test different time discretization steps.

**Keywords:** Large-scale nonlinear optimization, human-like collision-free arm movements

**PACS:** 02.60.Pn

## INTRODUCTION

In previous work we have presented a model for generating smooth and human-like arm and hand movements of an anthropomorphic robot, **ARoS**, that is inspired by the Posture-Based Motion-Planning Model (PBMP) of human reaching and grasping movements [1]. Several experiments have been made on human-robot interaction tasks (see [2, 3]). However, for a fluent and efficient human-robot interaction, movements must be generated in real-time or at least in an sufficiently small interval of time. In [4] we tested different three solvers for determining the solution of the large-scale nonlinear constrained optimization problems that rise from the discretization of our time-continuous model. Here we present some changes to the model presented in [4] and use other optimization techniques. Furthermore, we explore how the discretization steps affects the performance of the solvers both in terms of computational time and quality of the solution.

There is a large amount of available optimization software (see e.g. <http://www.neos-server.org/>), some of which are open-source (see e.g. <https://projects.coin-or.org/>). Roboticists have not played too much attention to it. An exception is [5], who use IPOPT [6] merely to solve the inverse kinematics problem of an anthropomorphic robotics arm in point-to-point movements in the absence of obstacles. In contrast, we modeled the entire trajectory of the arm and hand, which includes obstacle avoidance, in reaching and grasping movements. Additionally, several grip types are consider, specifically, side and above grips.

## THE MODEL

The model we have developed has been described in [4]. Here we give a brief description and present some changes.

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 degrees of freedom (DOFs). **ARoS**' anthropomorphic robotic arm has 7 DOFs and its hand has 4 DOFs. The arm and hand configuration in joint space is defined by the vector  $\theta = (\theta_1, \theta_2, \dots, \theta_{11})^\top$ . Taking inspiration from the PBMP by [1], we define the movement of each joint on the robotic arm and hand as the superimposition of two movements: (i) a direct movement, describing a bell-shaped unimodal velocity profile, from the initial to final posture; (ii) a *back-and-forth* movement from initial to a bounce posture, intended to avoid collision with obstacles in the robot's workspace. Therefore, the movement planning proposed here for the anthropomorphic robotic arm and hand can be summarised as the resolution of two subproblems: (Pa) determining the final posture, i.e.,

---

<sup>1</sup> Eliana Costa e Silva was supported by FCT (grant: SFRH/BD/23821/2005).

a vector of arm and hand joint angles,  $\theta_f \in \mathbb{R}^{11}$ , that allows **ARoS** to grasp a given object subject to specific constraints (e.g. grip type); (Pb) determining a bounce posture,  $\theta_b \in \mathbb{R}^{11}$ , that serves as a sub-goal for a *back-and-forth* movement. The sequence of joint angles of the robotics arm and hand is given by

$$\begin{aligned} \theta(t, \theta_f, \theta_b) = & \theta_0 + (\theta_f - \theta_0) \left( 10\tau^3 - 15\tau^4 + 6\tau^5 \right) + v_0 T \left( \tau - 6\tau^3 + 8\tau^4 - 3\tau^5 \right) \\ & + \frac{1}{2} a_0 T^2 \left( \tau^2 - 3\tau^3 + 3\tau^4 - \tau^5 \right) + (\theta_b - \theta_0) \sin^2(\pi \tau^\vartheta), \end{aligned} \quad (1)$$

where  $\theta_0, v_0, a_0 \in \mathbb{R}^{11}$  are constant vectors representing initial joint position, velocity and acceleration, respectively,  $T \in \mathbb{R}^+$  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. We discretize  $t \in [0, T]$  by  $N_T$  equally spaced points  $t_i = i\Delta$ , where  $\Delta = \frac{T}{N_T}$  is the step size and  $i = 0, 1, \dots, N_T$ . Our convention is that  $\theta(t_i, \theta_f, \theta_b)$  represents  $\theta(t, \theta_f, \theta_b)$  at time  $t_i$ . The two subproblems (Pa) and (Pb) are:

$$\text{(Pa) } \min_{\theta_f \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k (\theta_{0,k} - \theta_{f,k})^2, \lambda_k \geq 0 \quad (2)$$

$$\text{s.t.} \quad H(\theta_f) + d_{HO}(\theta_{f,9}) \hat{z}_7(\theta_f) - X_{tar} = 0 \quad (3)$$

$$c(\theta_f) = 0 \quad (4)$$

$$h_f(\theta_f) \leq 0 \quad (5)$$

$$\theta_{min} \leq \theta_f \leq \theta_{max} \quad (6)$$

$$\theta_{f,8} = 0, \theta_{f,9} = \theta_{f,10} = \theta_{f,11} \quad (7)$$

$$\text{(Pb) } \min_{\theta_b \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k (\theta_{0,k} - \theta_{b,k})^2, \lambda_k \geq 0 \quad (8)$$

$$\text{s.t.} \quad \theta_{min} \leq \theta(t_i, \theta_f, \theta_b) \leq \theta_{max} \quad (9)$$

$$\theta_8(t_i, \theta_f, \theta_b) = 0, \theta_9(t_i, \theta_f, \theta_b) = \theta_{10}(t_i, \theta_f, \theta_b) \quad (10)$$

$$h_b(\theta(t_i, \theta_f, \theta_b)) \leq 0, \quad t_i = 0, \dots, T \quad (11)$$

$$\theta_{min} \leq \theta_b \leq \theta_{max} \quad (12)$$

where  $\theta_{min}, \theta_{max}$  are constant vectors that represent the lower and upper mechanical joint limits,  $X_{tar} \in \mathbb{R}^3$  is the position of the object that the robot must grasp,  $d_{HO}(\theta_{f,9})$  is the distance from the center of the object to the palm of the hand. The equality constraints (4) depend on how one wants to grasp the object, i.e., the desired grip type. The fingers joint angles (7) are determined using object dimensions and grip type. The nonlinear inequality constraints defined by (5) and (11) are due to obstacle avoidance and were presented in [4]. The dimension of (Pa) depends on the number of constraints in (5), which depend on the number of obstacles in the workspace of the robot. This is, in general, a small-scale optimization problem (see Table 1). On the other hand, (Pb) is in general a large-scale problem (see Table 2), with the number of constraints in (11) depending on the number of obstacles and also on the time discretization step that is used. Here we have introduced some changes to the model presented in [4] intended to improve the modeling of reach to grasp movements. The first concerns constraints (4) relative to hand orientation. We have replaced this equality by the inequality constraint  $\|c(\theta_f)\|^2 \leq \delta$ ,  $\delta > 0$ . In fact, when grasping an object there is a higher priority in hand position (given by (3)) than on hand orientation, since it does not need to be met perfectly. The second change concerns the constraints in (11), that are due to obstacle avoidance. Here we distinguish collision of the robotic arm and hand with objects in the workspace of the robot and the collision with the object that is going to be grasped. Thus,  $h_b = (\underline{h}_b, \bar{h}_b)^\top$  where  $\bar{h}_b \equiv \bar{h}_b(\varepsilon(t))$  concerns the target object and  $\underline{h}_b$  concerns other objects in the workspace. The function  $\varepsilon(t)$  is a monotonously decreasing to zero function representing the clearance distance.

## RESULTS

Tables 1 and 2 report the results for four problems<sup>2</sup>, each consisting of two subproblems. These problems arise in tasks in which the anthropomorphic robot **ARoS** has to grasp objects using different grip types, and where the reach-to-grasp movement may be challenged by the presence of multiple obstacles. Specifically, we focus here on reaching and grasping columns and wheels, using a side and an above grip, respectively. For a definition of the grip types and on the details of the task see e.g. [2, 3]. Previously in [4] we have tested the performance of three nonlinear constrained optimization solvers (IPOPT, KNITRO and SNOPT) for these four problems using a fixed discretization of time and with slightly different constraints (see previous section). All problems are coded in AMPL (<http://www.ampl.com/>) modelling language and were submitted to the NEOS Server<sup>3</sup> platform. We report the results for two nonlinear constrained optimization solvers, IPOPT and SNOPT, and four mixed integer nonlinearly constrained optimization solvers, BONMIN, COUENNE, FilMINT and MINLP. The choice on these solvers has to do with the fact they use different optimization techniques. IPOPT is an open source large-scale optimization solver that implements an Interior Point (IP) filter line search method, while SNOPT implements an active-set Sequential Quadratic Programming (SQP) line search method. BONMIN is an open source software for the solution of mixed integer nonlinearly constrained optimization problems that presents two classical algorithms: an outer-approximation-based branch-and-cut-based algorithm and a pure branch-and-bound algorithm. COUENNE solves the same type of problems using a reformulation based branch-and-bound algorithm for a globally optimum solution. FilMINT combines the branch-and-cut framework with filterSQP, an active set solver, used to solve the nonlinear programs that arise as subproblems in the algorithm. MINLP implements a branch-and-bound algorithm and uses a filterSQP for solving the continuous nonlinearly constrained optimization problems that arise as nodes of a tree. All these solvers guarantee global optimal solutions only for convex mixed integer nonlinearly constrained optimization problems. Only COUENNE guarantees global optimal solutions for nonconvex ones. For P1a and P2a the constraint (4) is given by  $\|\hat{x}_7(\theta_f) - \hat{z}_{tar}\|^2 \leq \delta$ , while for P3a and P4a is  $\|\hat{z}_7(\theta_f) + \hat{z}_{tar}\|^2 \leq \delta$ , where  $\hat{x}_{tar}\hat{y}_{tar}\hat{z}_{tar}$  is the local frame attached to the object that the robot must grasp, and  $\hat{x}_7$  and  $\hat{z}_7$  are forward kinematics functions describing the orientation of the hand of the robot. In our implementation the value of the following constants are:  $T = 1$ ,  $t_b = 0.5$ ,  $\delta = 10^{-4}$  and  $\lambda_k = 1, k = 1, \dots, 11$ . All solvers were run with the default options, with the exception of IPOPT, for which the second order derivatives information were approximated using a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method. Table 1 presents the numerical results for the small-scale problems P1a-P4a. For all problems IPOPT, SNOPT, BONMIN and MINLP found an optimal solution. FilMINT only found an optimal solution for P1a and P2a. COUENNE reported error for all problems. For all problems BONMIN found the worse solution. All the other solvers found the same solution for P2a, P3a and P4a. IPOPT found the best solution for P1a. FilMINT and MINLP found the same solution as SNOPT, which is not the best one. This may be explained by the fact that, for P1a, and since they are mixed integer nonlinearly constrained optimization solvers, they have incorrectly identified one of the variables as integer. In fact, for this problem all solvers, except IPOPT, found  $\theta_6 = 0$ . For P2a, P3a and P4a, none of the variables is identified as integer. In terms of computational time, the best results were obtained by IPOPT and SNOPT.

**TABLE 1.** Numerical results for the small-scale problems P1a-P4a.

	<b>P1a</b> ( $M = 45$ )		<b>P2a</b> ( $M = 66$ )		<b>P3a</b> ( $M = 45$ )		<b>P4a</b> ( $M = 66$ )	
	Obj.	CPU(sec.)	Obj.	CPU(sec.)	Obj.	CPU(sec.)	Obj.	CPU(sec.)
IPOPT	4.579811	0.071	4.579811	0.067	6.763172	0.049	6.763172	0.152
SNOPT	5.913691	0.064	4.579814	0.631	6.763174	0.014	6.763176	0.017
Bonmin	13.213600	1.025	14.114600	0.887	23.646300	1.090	8.342650	0.347
FilMINT	5.913691	0.110	4.579814	0.170	†		†	
MINLP	5.913691	‡	4.579814	‡	6.763176	‡	6.763176	‡

† No feasible solution found. ‡ Not available.

$M$  - number of constraints; Obj. - objective function value; CPU - Computational time in seconds.

Table 2 reports the numerical results for the large-scale problems P1b-P4b for three different discretization steps ( $\Delta = 0.1, 0.05, 0.025$ ). As expected, the dimension of the problems, i.e., the number of constraints,  $M$ , and the CPU

<sup>2</sup> For the videos of all the trajectories see <http://mobileanthropomorphicroboticsgroup.blogspot.com/>.

<sup>3</sup> Note that, although we have access to the names of the machines used to solve the problems, we do not know their technical specifications.

**TABLE 2.** Numerical results for the large-scale problems P1b-P4b.

	<b>P1b</b>			<b>P2b</b>		
	$\Delta = 0.1$ ( $M = 579$ )	$\Delta = 0.05$ ( $M = 1199$ )	$\Delta = 0.025$ ( $M = 2439$ )	$\Delta = 0.1$ ( $M = 768$ )	$\Delta = 0.05$ ( $M = 1598$ )	$\Delta = 0.025$ ( $M = 3258$ )
	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)
IPOPT	1.586542/1.560	0.727876/2.315	1.507861/27.674	1.586514/7.896	1.030850/3.978	2.180370/12.36 <sup>#</sup>
SNOPT	0.727859/0.078	0.727859/0.174	1.240994/0.668	0.932572/0.108	0.933127/0.312	1.621998/2.258
<hr/>						
	<b>P3b</b>			<b>P4b</b>		
	$\Delta = 0.1$ ( $M = 579$ )	$\Delta = 0.05$ ( $M = 1199$ )	$\Delta = 0.025$ ( $M = 2439$ )	$\Delta = 0.1$ ( $M = 768$ )	$\Delta = 0.05$ ( $M = 1598$ )	$\Delta = 0.025$ ( $M = 3258$ )
	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)	Obj./CPU (sec.)
IPOPT	0.765325/0.776	0.767077/1.259	0.768133/2.838	0.921092/3.039 <sup>#</sup>	1.015694/5.015	1.44493/99.595 <sup>#</sup>
SNOPT	0.751773/0.061	0.754943/0.172	0.755340/0.422	0.834245/0.101	0.861077/0.236	0.837660/0.634

<sup>#</sup> Converged to a point of local infeasibility;  $M$  - number of constraints; Obj. - objective function value; CPU - CPU in seconds.

time increased as the time discretization step gets smaller. SNOPT presents a better performance both in terms of the value of the objective function and CPU time. In terms of the time step discretization, SNOPT presents the best results for  $\Delta = 0.1$ . This was not the case for IPOPT. In fact, for P1b and P2b, it presents the best results for a discretization of  $\Delta = 0.05$ . Furthermore, for these two problems and for  $\Delta = 0.1$ , the trajectory found by IPOPT, results in the collision of the fingers of the robot, between the last two steps of the movement, with the object it is grasping<sup>4</sup>. For almost all problems, the mixed integer nonlinearly constrained optimization solvers were not able to find a solution. BONMIN reported that the limit number of iterations has been reached for all problems except for P1b ( $\Delta = 0.025$ ), P2b ( $\Delta = 0.1$ ) and P4b ( $\Delta = 0.1$ ), in which the solution found was worse than the one found by IPOPT and SNOPT (23.460, 5.95432 and 506.927, respectively). COUENNE reported error for all problems. FilMINT and MINLP reported failure in memory allocation for all problems, except for P3b ( $\Delta = 0.05$ ), in which FilMINT found the same solution as IPOPT.

## CONCLUSIONS AND FUTURE WORK

We have solved the problem of generating human-like reaching and grasping movements of an anthropomorphic arm and hand in the presence of obstacles, using different time discretization steps and different optimization techniques. IPOPT and SNOPT found the solutions for almost all problems and took less computational time. In general different methods found different solutions. The discretization of time still poses a problem. If it is too small the dimension of the problem increases greatly. On the other hand, if it is too large, collisions may occur. We tested the global solver COUENNE, however it reported error. In the future we expect to do further tests with the global solver COUENNE and other global solvers that accept GAMS input.

## REFERENCES

1. D. Rosenbaum, R. Meulenbroek, J. Vaughan, and C. Jansen, Posture-based Motion planning: Applications to grasping, *Psychological Review* **108**, 709–734 (2001).
2. E. Bicho, L. Louro, N. Hipólito, and W. Erlhagen, “A dynamic field approach to goal inference and error monitoring for human-robot interaction,” in *AISB Convention 2009 on Adaptive & Emergent Behaviour & Complex Systems: proc. of the Int. Symposium on New Frontiers in Human-Robot Interaction*, edited by E. Dautenhahn, Edinburgh, Scotland, 2009, pp. 31–37.
3. E. Bicho, W. Erlhagen, L. Louro, and E. Costa e Silva, Neuro-cognitive mechanisms of decision making in joint action: a Human-Robot interaction study, *Human Movement Science* (January 3, 2011) doi:10.1016/j.humov.2010.08.012.
4. E. Costa e Silva, F. Costa, E. Bicho, and W. Erlhagen, “Nonlinear Optimization for Human-like Movements of a High Degree of Freedom Robotics Arm-hand System,” in *11th Int. Conf. on Computational Science and Applications (ICCSA 2011)*, edited by B. Murgante et al., Springer-Verlag, 2011, vol. 6784, Part III of *Lecture Notes in Computer Science*, pp. 327–342.
5. U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini, “An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots,” in *Int. Conf. on Intelligent Robots and Systems (IROS2010)*, 2010, pp. 1668–1674.
6. A. Wächter, and L. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* **106**, 25–57 (2007).

<sup>4</sup> See the videos in <http://mobileanthropomorphicroboticsgroup.blogspot.com/>.