Generating Human-like Movements on an Anthropomorphic Robot using an Interior Point Method

E. Costa e Silva*, J.P. Araújo†, D. Machado†, M.F. Costa**, W. Erlhagen** and E. Bicho†

*Centre ALGORITMI/Portucalense University/ESTGF-IPP, Portugal
†Centre ALGORITMI/Dept. of Industrial Electronics, University of Minho, Portugal
**Centre for Mathematics/Dept. of Mathematics and Applications, University of Minho, Portugal

Abstract. In previous work we have presented a model for generating human-like arm and hand movements on an anthropomorphic robot involved in human-robot collaboration tasks. This model was inspired by the Posture-Based Motion-Planning Model of human movements. Numerical results and simulations for reach-to-grasp movements with two different grip types have been presented previously. In this paper we extend our model in order to address the generation of more complex movement sequences which are challenged by scenarios cluttered with obstacles. The numerical results were obtained using the IPOPT solver, which was integrated in our MATLAB simulator of an anthropomorphic robot.

Keywords: Large-scale nonlinear optimization, IPOPT, human-like movements, anthropomorphic robot, human-robot collaboration
PACS: 02.60.Pn

INTRODUCTION

For achieving natural and efficient human-robot interaction and collaboration, human-like movements are essential, since it allows the human involved in the interaction to interpret the robot's movements. Inspired by the Posture-Based Motion-Planning Model (PBMP) of human movements [2], we have, in previous work [1], presented a model for generating human-like arm and hand movements on an anthropomorphic robot involved in human-robot collaboration tasks (see [3]). The previous results focused on reach-to-grasp movements with two different grip types [1]. Here we extend our model in order to address the generation of more complex movement sequences (see Figure 1) which are challenged by scenarios cluttered with obstacles.

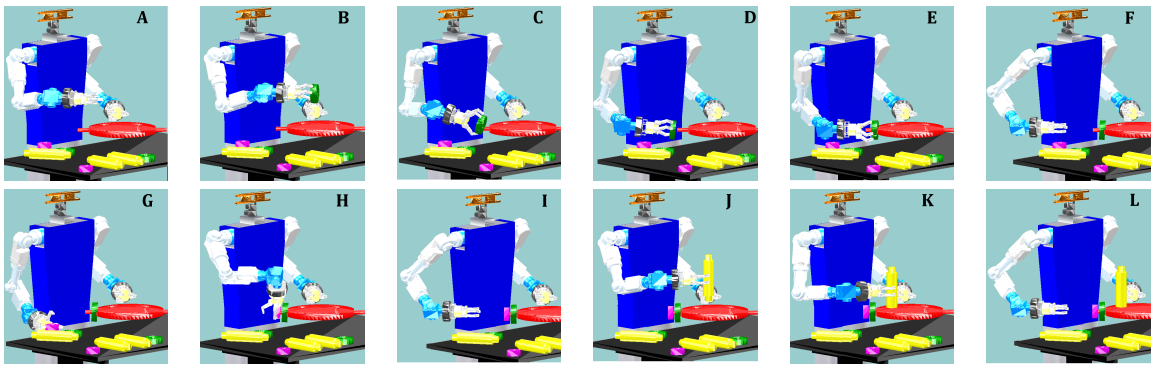

FIGURE 1. **ARoS** is an anthropomorphic robot equipped with two 7 degrees of freedom (DOFs) arms and two 4 DOFs hands. Panels A-F display snapshots, obtained in our MATLAB simulator, of a sequence of movements consisting of a request gesture (A), reach-to-grasp a wheel from the human (B), transporting and placing the wheel on the base (C-E), returning to the home position (F), grasping the nut from the table (G), inserting the nut (H-I) and a column (J-L) on the base.

Although the use of optimization in the generation of robot movements is not new (see e.g. [4]), roboticists have paid little attention to the large amount of available optimization software (see e.g. `https://projects.coin-or.org/`) and to the underlying optimization techniques. An exception is [5] where IPOPT [6] is used to solve the inverse kinematics problem of an anthropomorphic robotics arm in point-to-point movements in the absence of obstacles. In [7] IPOPT is used to determine the weights for achieving a better fit to observed human motion. Here we use IPOPT to solve

the nonlinear optimization problems that arise when we model the entire human-like trajectory of an anthropomorphic robot arm and hand, including obstacle avoidance. We use IPOPT because it is an open source software package for large-scale nonlinear optimization, that implements a primal-dual interior point method for solving nonlinear optimization problems. Here we further investigate the use of IPOPT in the real-time generation of more complex human-like movements.

THE MODEL FOR HUMAN-LIKE MOVEMENT ON AN ANTHROPOMORPHIC ROBOTIC ARM AND HAND

For the sake of brevity, some details of the model and the kinematics of the robotic system are omitted (for further details see [1]). The anthropomorphic redundant robotic arm and hand (Figure 1) can be represented as a series of links connected by joints. The number of joints which can be independently actuated define its 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 $\theta = (\theta_1, \theta_2, \ldots, \theta_{11})^\top$.

Taking inspiration from the PBMP [2], we define the movement of each joint 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 can be summarised as the resolution of two subproblems: **Pa** determining the appropriated final posture, i.e., a vector of arm and hand joint angles, $\theta_f \in \mathbb{R}^{n_j}$, that allows, for example, **ARoS** to grasp a given object or to achieve a specific location and grip type; **Pb** determining a bounce posture, $\theta_b \in \mathbb{R}^{n_j}$, that serves as a sub-goal for a *back-and-forth* movement. Here $7 \leq n_j \leq 11$ depends on the type of movement, as explained later. We formalize these problems as two nonlinear optimization problems with simple bounds and equality and inequality constraints.

The sequence of joint angles of the robotics arm and hand is given by

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

where $\theta_0, v_0, a_0 \in \mathbb{R}^{n_j}$ 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 $i = 0,1,\ldots,N_T$. Our convention is that $\theta(t_i,\theta_f,\theta_b)$ represents $\theta(t,\theta_f,\theta_b)$ at time t_i.

Here we simplified **Pa** and **Pb** as presented in [1]. First, since the middle finger is opposite to the other two we set $\theta_{f,8} = 0$. Second, the robotic hand has only one controllable DOF on each finger and all fingers have equal lengths, therefore for a successful grasp we have $\theta_{f,9} = \theta_{f,10} = \theta_{f,11}$. Thus, given the geometry of the hand, a specific object and grip type, the joint angles of the fingers $\theta_{f,9}$ are determined by solving, a transcendental equation, using Newton-Raphson method. Finally, we consider that during the movement $\theta_8(t_i) = 0$ and $\theta_9(t_i) = \theta_{10}(t_i), t_i = 0,\ldots,T$. Therefore **Pa** and **Pb** are:

$$\textbf{Pa}\ \min_{\theta_f \in \mathbb{R}^7} \quad \sum_{k=1}^{7} \lambda_k \left(\theta_{0,k} - \theta_{f,k}\right)^2, \lambda_k \geq 0 \tag{2}$$

$$\text{s.t.} \quad c_1(\theta_f) = 0 \tag{3}$$

$$\|c_2(\theta_f)\|^2 \leq \delta \tag{4}$$

$$h_f(\theta_f) \leq 0 \tag{5}$$

$$\theta_m \leq \theta_f \leq \theta_M \tag{6}$$

$$\textbf{Pb}\ \min_{\theta_b \in \mathbb{R}^{n_j}} \quad \sum_{k=1}^{n_j} \lambda_k \left(\theta_{0,k} - \theta_{b,k}\right)^2, \lambda_k \geq 0 \tag{7}$$

$$\text{s.t.} \quad \theta_m \leq \theta(t_i,\theta_f,\theta_b) \leq \theta_M \tag{8}$$

$$\underline{h}_b(\theta(t_i,\theta_f,\theta_b)) \leq 0 \tag{9}$$

$$\overline{h}_b(\theta(t_i,\theta_f,\theta_b),\varepsilon(t_i)) \leq 0, \tag{10}$$

$$\theta_m \leq \theta_b \leq \theta_M \tag{11}$$

$$t_i = 0,\ldots,T$$

where θ_m and θ_M are constant vectors that represent the lower and upper joint limits, $\delta > 0$ is a constant, c_1 and c_2 are nonlinear functions (of target pose (position and orientation) and joint angles, obtained using direct kinematics) concerning the position and orientation of the robot hand relatively to the target, respectively, $\varepsilon(t_i)$ is a function of

time representing the clearance distance, $h_f, \underline{h}_b, \overline{h}_b$ are nonlinear functions of the obstacles pose and of the arm and/or finger angles. For a more extensive description of the constraints see [1].

Request gestures consist of only **Pb** subproblems with $n_j = 9$ since we assume a fixed final pose of the arm and hand. Reach-to-grasp movements consist of one **Pa** and one **Pb** subproblems with $n_j = 9$. On the other hand, movements of transporting and placing an object do not allow movements of the fingers (since the robot is holding an object), thus for **Pb** $n_j = 7$. In this case the movement is composed of two sub-movements: the first from the initial posture to some location behind the insertion point; the second from this location to the insertion point (this is a direct movement). Therefore it consists of two **Pa** subproblems $\mathbf{Pa}_1$ for determining the pose of arm at the insertion point, and $\mathbf{Pa}_2$ for location behind the insertion point and one **Pb** subproblem.

RESULTS

We focus on a sequence of movements that arises in a human-robot joint construction task (see [3]). More specifically, we will report results on the following movements: **P1 ARoS** requests an object from the human; **P2** grasps the wheel that the human hands it; **P3** transports and inserts the wheel in the base; **P4** returns to its home position. Table 1 presents details on each of these movements.

TABLE 1. Problems description.

Problem	Movement	Target object	Other objects in the workspace	Final Posture Selection	Bounce Posture Selection
1	Request object	-	Table, base, wheel and column	-	**P1b**
2	Reach-to-grasp wheel	Wheel	Table, base and column	**P2a**	**P2b**
3	Transport and insert wheel	-	Table, base and column	$\mathbf{P3a}_1$ + $\mathbf{P3a}_2$	**P3b**
4	Return home	-	Table, base, column and wheel	**P4a**	**P4b**

All optimization problems, **P#a** and **P#b**, were coded in AMPL modeling language and solved using IPOPT 3.11. The numerical results were obtained using an Intel(R) Core(TM) i5 M460@2.53GHz processor running Windows 7 64-bits with a ATI Mobility Radeon HD 5650 video card and 4GB of Ram Memory. In our implementation the value of the following constants are: $T = 1$, $t_b = 0.5$, $\delta = 10^{-2}$ (for **P2a** $\delta = 10^{-3}$) and $\lambda_k = 1, k = 1, \ldots, n_j$. IPOPT was run with the default options, with the exception for which the second order derivatives information were approximated using a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method and we set AMPL presolve off.

Tables 2 and 3 show the numerical results. In these tables we present the number of variables, N, equality constraints, M_{eq}, inequality constraints, M_{ineq}, the number of non-zero elements in equality constraint Jacobian, n_{eq}, and on the inequality constraint Jacobian, n_{ineq}, the objective function value, Obj, and the computational time in seconds, CPU.

TABLE 2. Numerical results for **Pa** subproblems.

	P2a	$\mathbf{P3a}_1$	$\mathbf{P3a}_2$	**P4a**
N	7	7	7	7
M_{eq}	3	3	3	3
M_{ineq}	75	105	126	126
n_{eq}	17	17	17	17
n_{ineq}	439	649	778	778
Obj	0.2465	0.7525	0.0495	0.0719
CPU	0.156	0.203	0.171	0.312

IPOPT managed to find an optimal solution for all problems, although some tuning was necessary in the definition of $\varepsilon(t)$. All **Pa** subproblems are small-scale optimization problems. Their dimension depends mainly on the number of constraints in (5), which depend on the number of obstacles in the workspace of the robot. IPOPT was able to find easily an optimal solution in less then 0.320 seconds. As for the **Pb** subproblems, their dimension is related to the number of constraints in (9) and (10), which depend on the number of obstacles, on the time discretization steps that are used and on the number of points considered on the robot's arm and hand (for further details see [1]). These subproblems are medium-scale ones and IPOPT found an optimal solution in less than 1.2 seconds. The movement of inserting and placing the wheel on the base was the one that presented the greater risk of collision with the surrounding

TABLE 3. Numerical results for **Pb** subproblems.

	P1b	**P2b**	**P3b**	**P4b**
N	9	9	7	9
N_T	10	10	20	10
M_{ineq}	711	906	786	531
n_{ineq}	3660	4900	2920	2425
Obj	3.663e-013	1.338e-02	1.548e+0	1.271e-015
CPU	0.858	0.749	1.123	0.764

obstacles. Therefore it was not surprising that **P3b** was the most challenging subproblem for the IPOPT solver, as can be seen by the CPU time taken. Also for this movement the bounce component, responsible for avoiding collision, was greater then for all the other movements. This is expressed by the value of the objective function.

CONCLUSIONS AND FUTURE WORK

We have used IPOPT integrated in our system to solve the nonlinear optimization problems that arise when we model the entire human-like trajectory of an anthropomorphic robot arm and hand, including obstacle avoidance. We further have demonstrated that, even with modest computational resources, the CPU times obtained are appropriated for real-time implementation since they allow the human interacting with the robot to interpret its movements.

In the future we expect to perform user studies and extend our approach to bimanual manipulation of the anthropomorphic robot **ARoS**.

ACKNOWLEDGMENTS

This work was financed by EU funded Project PF7 Marie Curie "NETT - Neural Engineering Transformative Technologies", by FEDER funds through COMPETE (Operational Programme Thematic Factors of Competitiveness) and by portuguese funds through FCT (Foundation for Science and Technology) within the projects PEst-C/MAT/UI0013/2011 and FCOMP-01-0124-FEDER-022674.

REFERENCES

1. 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.
2. D. Rosenbaum, R. Meulenbroek, J. Vaugham, and C. Jansen, Posture-based Motion planning: Applications to grasping, *Psychological Review* **108**, 709–734 (2001).
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. N. Ratliff, M. Zucker, J.A. Bagnell, and S. Srinivasa, "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning", in *IEEE Int. Conf. on Robotics and Automation (ICRA2009)*, 2009.
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).
7. S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," in *11th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids2011)*, 2011, pp. 602–607.