Motion Optimization of Robotic Systems and Validation on HRP-2 Robot

Sylvain MIOSSEC (CNRS/AIST), Sebastien LENGAGNE (CNRS/LIRMM) Abderrahmane KHEDDAR (CNRS/AIST), Kazuhito YOKOI (AIST)

Abstract— This paper presents a software devised for the purpose of optimal motions computation for robots. Generated motions satisfy a stability constraint and motors limitations while minimizing energy consumption. Motion optimization is solved with the IPOPT optimization package. In order to achieve fast and reliable optimization convergence, calculation of gradients is carried out. Almost rarely considered in the literature, joint friction, which has a non-negligible effect on the optimized motion, is taken into account. Efficiency of the proposed software is demonstrated through the optimization for the 30 degrees of freedom humanoid robot HRP-2 of three dynamic motions: a kicking motion, a throwing motion and a lifting motion. We have applied successfully the obtained optimal motions to the real robot. We show that the motion optimization approach is a powerful tool to generate various types of motions, and to take into account nonlinear limitations.

Key Words: Optimal motion, humanoid, joint friction.

1. Introduction

Considerable work has been devoted to optimal control of systems. This is due to the fact that in many fields, e.g. aerospace, it is important to obtain energy or time optimal motions. Optimization problems can be solved with (i) indirect methods using the Pontryagin's Maximum Principle, and (ii) direct methods that consist in solving the discretized problem by using parametric optimization techniques. Direct methods are easier to use, have a larger convergence domain but are less precise than indirect methods. Direct methods can be classified in three approaches: (i) Collocation methods, see for example [1], (ii) Multiple-shooting methods, see for example [2] (iii) Methods based on parameterization of the state variables, and obtaining the control variables from the inverse dynamic model. [3], [4] and [5] have proved that this last method is more efficient for fully actuated robots.

Optimization techniques has been used to optimize (virtual) human motions as in [6] and [7]. In robotics, optimal motion generation has been used for the problem of walking of simple robots in [8] and [9]. In [10] a database of 3D optimal motions has been generated for a 13-dof walking robot but not experimentaly used. In [11] motions for closed-chained mechanical systems have been optimized, but no constraints where considered. None of these works considered joint's friction, which however accounts for a large part of joint torques. Recently, an interesting result shows that the regularization of static friction allows to solve the problem with usual techniques, provided the discretization of the problem is sufficient [12].

Optimal motion generation for robots is not used very often due to the complexity, heavy computations of the method and the absence of easy-to-use dedicated softwares. In this paper we present a general software that can be used for different type of motions and robots. We improved the efficiency of the gradient computation of the dynamic model, as proposed in [6] and [13] and [11], by considering the dependences in the recursive Newton-Euler dynamics computations. We also considered the joint friction in the motion optimization. For this, we used a regularization method. Section 2. presents the general motion optimization problem for fully actuated robots. Section 3. presents our software, the constraints formulation, and the gradients computation method. Section 4. exemplifies our software on three different motions for the HRP-2 robot and experimental applications. Finally, we give our conclusions and perspectives in section 5..

2. Problem statement

The exact motion optimization problem to solve is the following

$$\min_{q(t),u(t),F(t),t_f} \mathcal{C}(q(t),\dot{q}(t),u(t),t_f) \tag{1a}$$

subject to

$$u = f(q,\dot{q},\ddot{q},F) \tag{1b}$$

$$c_{\text{meq}}(q,\dot{q},\ddot{q},u) = 0 \tag{1c}$$

$$c_{\text{mineq}}(q,\dot{q},\ddot{q},u) \leq 0 \tag{1d}$$

$$c_{\text{mt}}(q(t_d),\dot{q}(t_d),\ddot{q}(t_d),u(t_d)) = 0 \tag{1e}$$

where q is the vector of parameters of the system, u is the control vector and F the vector of forces and torques applied by the environment, (1b) is the dynamic model of the system, (1c) and (1d) are the constraints at every instant of the motion and (1e) constraints at fixed instants. The goal is to find the optimal functions $q(t)$, $u(t)$, $F(t)$, and the final time t_f for which the criteria $\mathcal{C}$ is minimized and constraints are satisfied.

3. Presentation of the software

To solve (1) Our software implements a direct method that consists in the discretization of joint angles j and forces k as B-splines (2), and motion constraints (1c) and (1d) at discreet points during the motion.

$$\begin{array}{l} q_j(p,t) = \sum_{i=1}^{n_{qj}} B_{q,i}(t) c_{q,ij} \\ F_k(p,t) = \sum_{i=1}^{n_{Fj}} B_{F,i}(t) c_{F,ik} \end{array} \qquad (2)$$

where n_j is the number of basis functions $B_i(t)$, c_{ij} are the B-spline coefficients, N_q is the total number of joints, N_F is the total number of force components. The parameters of the motion are then $p = \{c_{q,ij} \mid j \in [1, N_q], i \in [1, n_{qj}]\} \bigcup \{c_{F,ik} \mid k \in [1, N_F], i \in [1, n_{Fj}]\}$..

Additionaly to B-spline computations, our software includes the definition of the motion characteristics, the dynamic computations, the constraints definition and computation, the criteria computation. Gradients are also computed. The software includes an interface with the optimization program IPOPT (see [14] for more details).

3·1 Considered systems and model

We consider kinematic chains composed of revolute joints each of which having viscous and dry friction. We only address systems having at least one fixed contact with the environment of unilateral surface-surface type or bilateral type.

The dynamics of the considered class of system can be modeled as follows,

$$u_m = A(q)\ddot{q} + H(q,\dot{q}) - J^T F \qquad (3)$$

$u_m \in \mathbb{R}^n$ is the joint torque applied to the mechanical system, $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, $\ddot{q} \in \mathbb{R}^n$ are respectively the joints' positions, velocities and accelerations, $A(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $H(q,\dot{q}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravity effects, F is the vector of external forces and torques applied on the robot, and J^T is the transpose of the Jacobian between the space of application of F and the joint space. This dynamic model is computed with the recursive Newton-Euler algorithm.

The joint friction are included by considering the control input being u, the joint torque u_m with the effects of friction, that is

$$u_j = u_{m,j} + \frac{2u_{d,j}}{\pi} \arctan(\dot{q}_j c_r) + u_{v,j}\dot{q}_j \qquad (4)$$

u_d is the torque of static friction, c_r is the regularization coefficient for the static friction, u_v is the coefficient for viscous friction. The bigger c_r is, the better the static friction approximation, but also the stiffer the system is.

3·2 Criteria considered

For the time being, we implemented an energy criterion that accounts for mechanical losses due to friction, and the losses in motors resistors,

$$\mathcal{C}(\dot{q}, u, t_f) = \int_{t_0}^{t_f} \sum_{j=1}^{N} \frac{R_j {u_j}^2}{K_{em,j}^2} + u_j \dot{q}_j \, dt \qquad (5)$$

where R_j is the resistance and $K_{em,j}$ the electromechanical constant for the actuator of joint j. Other useful criteria can be easily programmed, like minimize the time or maximize the speed of a body.

3·3 Constraints

Several constraints can be added to the system. They can be divided into two types: (i) physical limitations of the system and (ii) motion constraints defining the characteristics of the desired motion.

The physical limitations are expressed as follows:

- joint limits $q_{\min} \le q \le q_{\max}$.
- the main actuators limits are the maximum supply voltage that gives $-u_{\max} \le u + \frac{u_{\max}}{\dot{q}_{\max}}\dot{q} \le u_{\max}$ and the maximum speed $-\dot{q}_{\max} \le \dot{q} \le \dot{q}_{\max}$.
- constraints of no sliding, no take off, no turn over the edges of the contact of a body are given by

$$\begin{cases} \sqrt{f_x^2 + f_y^2} - \mu f_z \le 0 \\ -f_z x_{\text{neg}} \le f_x z_{\text{surf}} + m_y \le f_z x_{\text{pos}} \\ -f_z y_{\text{neg}} \le f_y z_{\text{surf}} - m_x \le f_z y_{\text{pos}} \end{cases} \qquad (6)$$

  In the contact frame, $f_{x,y}$ and f_z are respectively the tangential and normal components of the external forces acting on the base, $m_{x,y}$ are the external moments acting on the base, x_{neg}, x_{pos}, y_{neg}, y_{pos} are the edges of the contact surface respectively in x and y directions, and in negative and positive directions, and z_{surf} is the distance along normal axis between the surface and the origin of the body. We did not considered the friction constraint in rotation.

The constraints on the motion that are considered:

- position constraints of body j, at some discrete instant t_d of the motion, given by $\text{pos}_j(q(t_d)) = \text{pos}_{j,d}^0$.
- inequality constraints on the position of body j of the system, for all discrete instants t_k, are given by $n.\text{pos}_j(q(t_k)) \le n.\text{OP}^0_{\text{front},j}$ where n is the normal to the plane delimiting the admissible zone for the origin of the body j, $\text{OP}^0_{\text{front},j}$ is a vector from the origin of absolute frame to a point belonging to this plane, and symbol '.' is the scalar product.
- inequality constraint on the velocity of body j in direction n, at discrete instants t_d, is given by $n.\text{vel}_j(q(t_d)) \ge \text{vel}^0$.

- orientation constraint of body j, at discrete instant t_d is given by

$$\begin{cases} \theta_1(q(t_d)) = \arctan\left(\frac{u_{j,x_jz_j}.z_j}{u_{j,x_jz_j}.x_j}\right) = 0 \\ \theta_2(q(t_d)) = \arctan\left(\frac{u_{j,x_jy_j}.y_j}{u_{j,x_jy_j}.x_j}\right) = 0 \\ \theta_3(q(t_d)) = \arctan\left(\frac{v_{j,y_jz_j}.z_j}{v_{j,y_jz_j}.y_j}\right) = 0 \end{cases} \quad (7)$$

  where (u_j, v_j, w_j) is the frame of body j, (x_j, y_j, z_j) is the absolute frame to which (u_j, v_j, w_j) superposes, the symbol '.' is the scalar product, and u_{j,x_jz_j} is the projection of u_j in the plane defined by vectors x_j and z_j. This way, the orientation is simple to compute, avoiding singularities and behaving well during the optimization process. It also has a unique solution.
- fixed joint angle j for all discrete instants t_k is given by $q_j(t_k) = q_j^0$.

3·4 Gradient of dynamic model

For the efficiency of the optimization process, it is recommended to compute analytically and efficiently the criterion and the constraints gradients with respect to the optimization parameters p. The main computation occurs with the gradient of joint torques $\frac{\partial u_k}{\partial p}$. Such computations can be found in references [6, 13, 11]. An improvement of our gradient computation comes from the consideration of the system structure which allows computing only the non-zero components of the gradient, while keeping a generic algorithm. For this, we analyzed the dependencies of the recursive Newton-Euler computations. More details can be found in [15].

4. Results

The proposed software has been used for the generation of optimal trajectories for the Humanoid robot HRP-2 [16]. We used a dynamic model of the robot including experimentally identified viscous and static friction. The dynamic parameters were obtained from the CAD model of the robot.

4·1 Motion optimization

For all motions, we imposed some of the constraints presented in section 3.. We considered joints limits, actuators limits, no sliding, no take off, and no turn over of the contacts. We considered the following caracteristics for each motion:

- **Kicking motion**. We specified initial and final equality constraints on the position of the free foot. To deal with auto-collision we used inequality constraints on the position of the hands and elbows. Those inequality constraints were tuned in order to obtain enough clearance without overconstraining the motion. In order to implement the motion on the real robot, we also considered a 20 degrees constraint on the knee of the supporting leg. Indeed, the controller implemented on the robot that corrects the effects of flexibility in the feet and control the ZMP position, does not deal with stretched legs.
- **Throwing motion**. The initial and final configuration are fixed. The only constrained used to define the motion is a constraint on the speed of the hand at the middle of the motion.
- **Lifting motion**. An object of 8kg is considered. Weight of the object is obtained by artificially increasing the weigth of the hands. Stable initial and final configurations are pre-computed. The motion is restricted to the sagittal plane by constraining some angles to 0deg. More details are given in the paper [17].

For the parameterization of the kicking (throwing, lifting) motion, we choose $n_{qj} = n_{Fj} = 9(15)$ B-spline basis functions. For the computation of the criterion we used 61(121) integration points and a trapezoidal integration scheme. For the constraints, we considered 13(25) discrete points.

All optimization times are about 10min. The obtained kicking motion duration is 809ms, the throwing motion duration is 884ms, and the lifting motion duration is 1.645s, which is fast motions for humanoid robots.

4·2 Experimental results

We have implemented the optimized motions with the flexibility stabilizer. We obtained successfully stable motions. The snapshots of the experiments are presented in fig. 1. We have also experimented a lifting motion without actuator constraints. It could not be executed by the real robot, which shows the importance of taking into account actuators limitations.

5. Conclusion

In this paper, we have devised dedicated software for the optimization of motions for robots. We used an efficient way to compute the gradient of the dynamics with respect to optimization parameters. We have shown the efficiency of our algorithm for three dynamic motions. We have also taken into account the joint dry friction in order to obtain optimal motions closer to real optimal motions of robots. Those dynamic motions have been successfully implemented on the real robot.

[1] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, pp. 338–342, 1987.

[2] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proc. 9th IFAC Worl Congress*, vol. IX, 1984, pp. 242–247.

[3] M. C. Steinbach, "Optimal motion design using inverse dynamics," Tech. Rep., 1997.

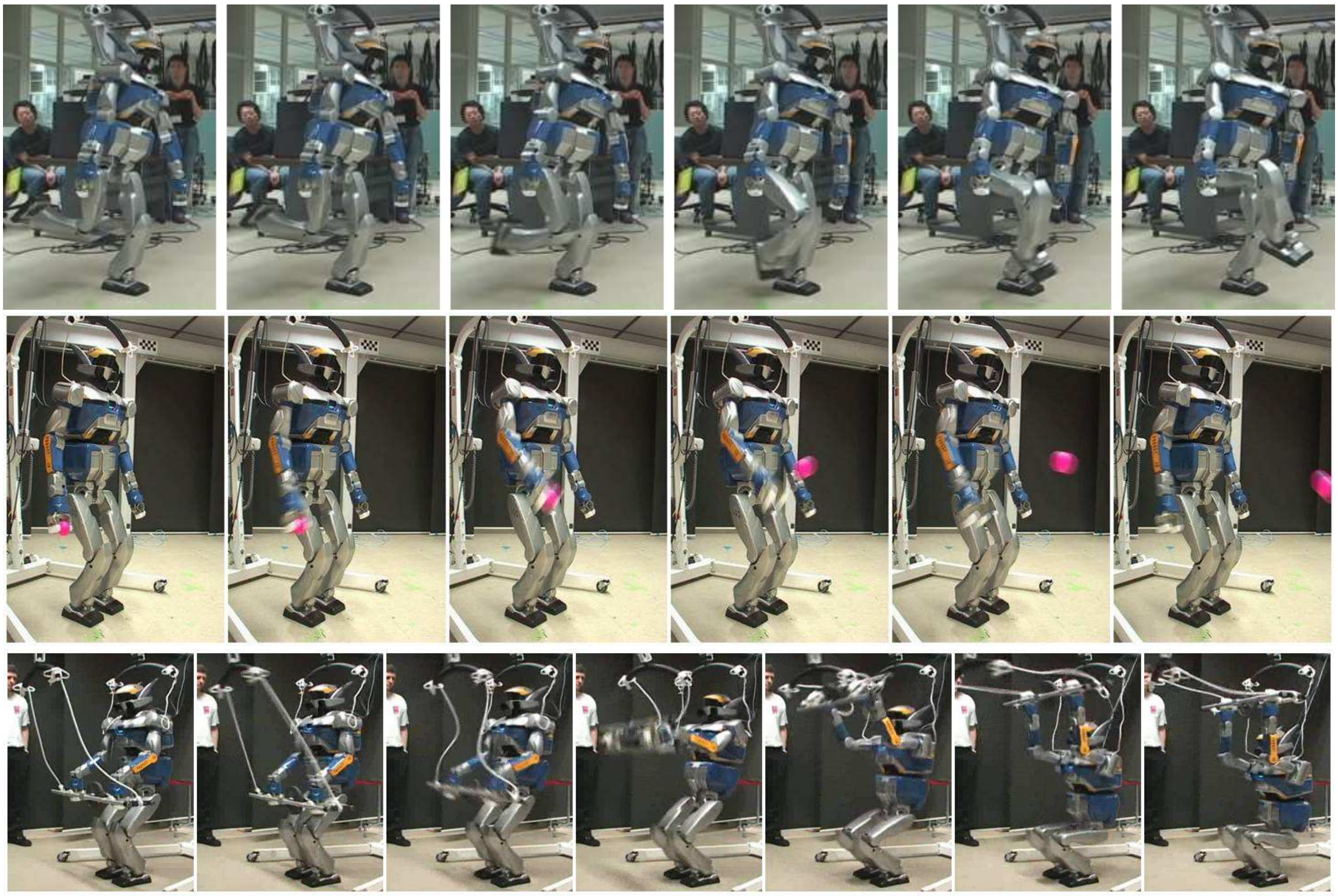

Fig.1 Optimized kicking motion (top), throwing motion (middle) and lifting motion (bottom).

[4] O. Von Stryk, "Optimal control of multibody systems in minimal coordinates," in *Proceedings of the Annual GAMM Conference*, 1997.

[5] R. Kumar and H. Seywald, "Should controls be eliminated while solving optimal control problems via direct methods?" *Journal of Guidance, Control and Dynamics*, vol. 19, no. 2, pp. 418–423, March-April 1996.

[6] J. Lo, G. Huang, and D. Metaxas, "Human motion planning based on recursive dynamics and optimal control techniques," *Multiboldy System Dynamics*, vol. 8, pp. 433–458, 2002.

[7] A. Safonova, J. Hodgins, and N. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 514–521, August 2004.

[8] M. Hardt, K. Kreutz-Delgado, and J. W. Helton, "Optimal biped walking with a complete dynamical model," in *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999.

[9] C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica*, vol. 19, pp. 557–569, 2001.

[10] J. Denk and G. Schmidt, "Synthesis of a walking primitive database for a humanoid robot using optimal control techniques," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2001, pp. 319–326.

[11] S.-H. Lee, J. Kim, F. Park, M. Kim, and J. Bobrow, "Newton-type algorithms for dynamics-based robot movement optimization," *IEEE Transactions on Robotics*, vol. 21, no. 4, August 2005.

[12] D. E. Stewart and M. Anitescu, "Optimal control of systems with discontinuous differential equations," *submitted*, 2006.

[13] G. Sohl, "Optimal dynamic motion planning for underactuated robots," Ph.D. dissertation, University of California, 2000.

[14] A. Wachter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[15] S. Miossec, K. Yokoi, and A. Kheddar, "Development of a software for motion optimization of robots - application to the kick motion of the hrp-2 robot," in *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics*, Kunming, 2006.

[16] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, K. Hirata, M. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004.

[17] H. Arisumi, S. Miossec, J.-R. Chardonnet, and K. Yokoi, "Dynamic lifting by whole body motion of humanoid robots," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.