Multi-Contact Motion Generation: Continuous Constraints and Contact Forces

Sébastien LENGAGNE, Paul Mathieu, Abderrahmane KHEDDAR, Eiichi YOSHIDA

Abstract— This paper proposes a new computation method to plan multi-contact motions. The first problem is to deal with continuous equality and inequality constraint which rely on the physical limits of the robot and on the properties of the desired motion. The second problem in the modeling of a multi-contact motion is the indetermination relative to the contact forces. We propose a method to compute them from the contact stance and form the joint trajectories.

Key Words: Taylor Polynomials, continuous constraints, contact forces.

Introduction

We define a multi-contact motion a motion without any flying phase, i.e. the robot has at least one contact with the environment. Thus, most of the motion, (walking, climbing, sitting, ...) are multi-contact motions. For such motions, walking for instance, some methods based on simple model give efficient results [1], but if we want not to consider any assumption about the model, or if we aim at planning more general motions, we need a method that considers the full-body model of the robot and takes into account all its limits.

The multi-contact motion planning problem is to find the best joint trajectories that minimize a cost function and ensure a set of constraints. By considering a joint trajectory parametrization through Bsplines, the motion planning problem turns into finding the best parameters $\mathbf{P}$ that:

$$\begin{array}{ll} min & C(\mathbf{P}) \\ \forall i, \forall t \in [\Delta_i] & g_i(\mathbf{P}, t) \leq 0 \\ \forall j, \forall t \in [\Delta_j] & h_j(\mathbf{P}, t) = 0 \end{array} \tag{1}$$

With $C(\mathbf{P})$ the cost function, g_i, the set of continuous inequality constraint ensure the physical limits of the robot such as joint position velocity and torque limits and h_i the set of the continuous equality constraints that produces a geometrical constant position of one body when it is in contact with the environment.

1. Dealing with constraints

In [2], we show that the joint position and velocity all along the motion are ensured by only considering constraint on the parameters. In fact, a Bsplines function is entirely contained in its control point convex hull. Hence, by limiting the control point to the maximal joint value, the joint trajectory will not violate the joints limits. Moreover, since the derivative of a Bsplines functions is another one, we can use the same way to constraint the joint velocity.

For more complex constraints, such as the joint torques, we cannot use this method. We approximate all the variables of the system by a n-order polynomial (those values are computed through a C++-template model function):

$$f(t) \approx [a_0, a_1, \ldots, a_n] \times [1, t, \ldots, t^n]^T \tag{2}$$

We can easily compute a conservative value of the extrema by using the convex hull property of the Bsplines. We identify the equivalent Bsplines parameter:

$$f(t) = [p_0, p_1, \ldots, p_n] \times \mathbf{B} \times [1, t, \ldots, t^n]^T \tag{3}$$

where $\mathbf{B}$ is a matrix that contains the polynomial parameters of the B-Splines basis functions $b_i(t)$. (Note that the basis functions used to evaluate extrema are different from the basis function used to define the trajectories). We compute the corresponding B-Splines parameters such that:

$$[p_0, p_1, \ldots, p_n] = [a_0, a_1, \ldots, a_n] \times \mathbf{B}^{-1} \tag{4}$$

The Matrix $\mathbf{B}$ relies on the order of the Taylor approximation and on the time interval $[\Delta t]$, thus we need to compute it only once at the beginning of the optimization process. In addition, we are able to compute a conservative value of the inferior and superior bounds for any function thanks to:

$$\forall t \in [\Delta t] \quad \min_i(p_i) \leq f(t) \leq \max_i(p_i) \tag{5}$$

Moreover, thanks to the polynomial approximation we can easily compute the derivative of any variables and also constraint them to a constant value C^{st} by considering the following constraints:

$$\begin{array}{ll} \forall i \in \{1, \cdots, N_e\} & a_i = 0 \\ & a_0 = C^{st} \end{array} \tag{6}$$

2. Modeling contact forces

We define the contact forces as a set of linear forces attached to each contact body. The contact forces are explicitly used in the inverse dynamic model:

$$\begin{bmatrix} \Gamma \\ 0 \end{bmatrix} = \begin{bmatrix} M_1(q) \\ M_2(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} H_1(q, \dot{q}) \\ H_2(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} J_1^T(q) \\ J_2^T(q) \end{bmatrix} F_c \tag{7}$$

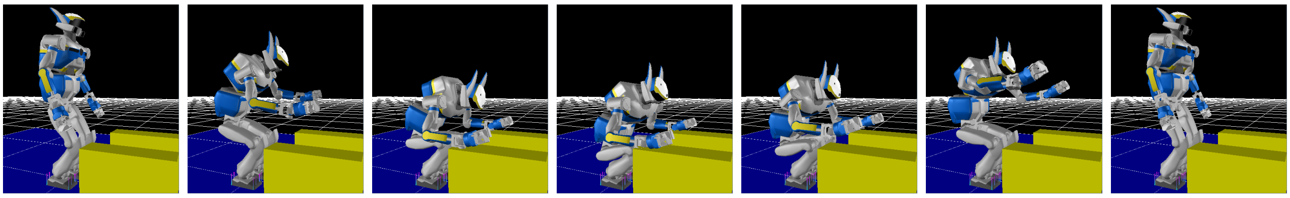

Fig.1 Flexing motion

To keep the balance of the robot the second line of Eq. 7 must be satisfied and each linear contact forces must ensure the unilaterality and no-sliding constraints:

$$\forall i, \forall t \in [\Delta t] \quad \begin{cases} F_i^n(t) > 0 \\ |F_i^t(t)|^2 \leq \mu_i F_i^n(t)^2 \end{cases} \tag{8}$$

We propose to find an analytical solution of the contact forces that solve the following problem:

$$\begin{gathered} \min \tfrac{1}{2} \textstyle\sum_i \beta_i (\alpha_i {F_i^t}^2 + {F_i^n}^2) \\ \textstyle\sum_i \left(\begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} [F_i] \right) + [D_2] = 0 \end{gathered} \tag{9}$$

With $\hat{P}_i$ is the screw operator of the contact position, $[D_2] = [M_x, M_y, M_z, F_x, F_y, F_z]^T$ the effort due to the free dynamics, β_i a coefficient to equilibrate (or not) the repartition of the forces and α_i that is used to weight the tangential effort regarding the normal force, to get forces as close as possible to the normal direction of the contact.

To get a solution of this problem, we start by writing the Lagrangian equation:

$$\begin{aligned} L = \quad & \textstyle\sum_i \beta_i \alpha_i ({F_i^t}^2) + \sum_i \beta_i {F_i^n}^2 \\ & + \left(\textstyle\sum_i \begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} [F_i] + [D_2] \right) [\lambda] \end{aligned} \tag{10}$$

Where $[\lambda]$ is the vector of the Lagrangian multiplicator. The optimal solution respects the condition of optimality :

$$\frac{\partial L}{\partial (F_{\circ,i}, \lambda_j)} = 0 \tag{11}$$

From the derivative with respect to the forces $F_\circ$ ($\circ = \{x, y, z\}$) we get:

$$F_i = -\gamma_i \begin{bmatrix} \hat{P}_i A_i & A_i \end{bmatrix} [\lambda] \tag{12}$$

With γ is a three component vector for which $\gamma_i = \frac{1}{\beta_i \alpha_i}$ if $F_\circ$ is one of the tangential component and $\gamma_i = \frac{1}{\beta_i}$ if $F_\circ$ is the normal component of the contact force. Then, we replace the contact forces expression in the set of the equality constraint and we get:

$$\sum_i \left(\begin{bmatrix} \hat{P}_i A_i \\ A_i \end{bmatrix} \gamma_i \begin{bmatrix} \hat{P}_i A_i & A_i \end{bmatrix} \right) [\lambda] = -[D_2] \tag{13}$$

We can transform this equation to have

$$\Omega [\lambda] = -[D_2] \tag{14}$$

Where Ω is a 6×6 matrix that we easily invert to find the value of the Lagrangian multiplicators:

$$[\lambda] = -\Omega^{-1} [D_2] \tag{15}$$

Then, we put the Lagrangian multiplicator value in Eq. (12) to get the value of the contact forces. We notice that if we define $\forall i \ \beta_i = 1$ and $\alpha_i = 1$, we solve the pseudo-inverse problem.

We notice that if we define $\forall i \ \beta_i = 1$ and $\alpha_i = 1$, we solve the pseudo-inverse problem.

3. Experiments

In [3], we present some results about the planning of multi-contact motion in 2D. This paper presents the first results about a 3D multi-contact motion with HRP-2 Robot. The robot is standing in front of an obstacle, he has to lean both arms on it and to go back to the initial position as presented in Figure 1.

Conclusion

We can generate dynamic multi-contact motion, since we can deal with continuous equality and inequality constraints and compute the contact forces, based on a dynamic full-body model of the robot. The next steps will be to implement for the collision avoidance in the optimization and to take into account the flexibilities of the actual robot.

[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2, september 2003, pp. 1620 – 1626.

[2] S. Lengagne, P. Mathieu, A. Kheddar, and E. Yoshida, "Generation of dynamic motions under continuous constraints: Efficient computation using b-splines and taylor polynomials," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[3] ——, "Generation of dynamic multi-contact motions: 2d case studies," in *IEEE-RAS Internationnal conference on Humanoid robots (submitted)*, 2010, pp. 105–110.