On the Dynamics Modeling of Free-Floating-Base Articulated Mechanisms and Applications to Humanoid Whole-Body Dynamics and Control
Karim Bouyarmane, Abderrahmane Kheddar

To cite this version:
On the Dynamics Modeling of Free-Floating-Base Articulated Mechanisms and Applications to Humanoid Whole-Body Dynamics and Control

Karim Bouyarmane* and Abderrahmane Kheddar†

*ATR Computational Neuroscience Laboratories, Kyoto, Japan
Email: karim.bouyarmane at atr.jp

†CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan
CNRS-UM2 Laboratoire d’Informatique de Robotique et de Micorélectronique de Monteppllier (LIRMM), France
Email: abderrahmane.kheddar at lirmm.fr

Abstract—We propose in this paper a general analytic scheme based on Gauss principle of least constraint for the derivation of the Lagrangian dynamics equation of motion of arbitrarily parameterized free-floating-base articulated mechanisms. The free-floating base of the mechanism is a non-actuated rigid object evolving in the 6D Lie group SE(3), the SO(3) component of which can be parameterized using arbitrary coordinate charts with equality constraints, for instance unit quaternions (also known as Euler parameters). This class of systems includes humanoid robots, and the presented formalism is particularly suitable for the whole-body dynamics modeling and control problem of such humanoid systems. Example motions of humanoid in arbitrary contact states with the environment demonstrate the originality of the approach.

I. INTRODUCTION

In [1] an elegant scheme for the derivation of the equation of motion of a humanoid robot is presented. This scheme allows for an analytic expression of the Lagrangian dynamics of humanoid robots, and more generally of any free-floating-base articulated mechanism (such as those studied in space robotics [2]–[5] before being applied to humanoids [6]), that contrasts with other algorithmically-oriented Newton-Euler recursive schemes, e.g. [7]. The analytic scheme directly considers the whole humanoid system, including its free-floating base, as one Lagrangian system with generalized coordinates that are simply time-differentiated once and twice in order to yield the equation of motion, rather than using specific kinematic quantities such as the angular velocity vector and its time derivative, that are not obtained through the time-differentiation of the generalized coordinates of the systems [8] [9] [10].

Advantages of this analytic approach are multiple. First it allows us to model the entire humanoid system without pre-specification of a particular contact state. The same model can thus be used for a humanoid in single-support stance on either foot, in double-support stance on both feet, or in any other stance on hands, knees, or elbows, with or without using the feet for support, see Fig. 1. It also allows for simple-to-implement yet general dynamics model algorithm provided that a sub-algorithm for the computation of basic geometric Jacobian matrices between two bodies of a kinematic chain is available, see [11] [12]. The dynamics model algorithm here just consists in the straightforward implementation of the presented analytic expressions. Finally, as opposed to classical treatments of Lagrangian dynamics, and as stressed by the author in [1], the proposed approach does not require the computation of the Christoffel symbols that need a powerful symbolic (algebraic) computation framework in order to be general enough to encompass more than one particular robot model.

![Fig. 1. Examples of fixed-base mechanism and free-floating-base mechanisms in various contact configurations.](image-url)
Jacobian matrices and push-forward mappings in Section II. We then apply Gauss principle to derive the equation of motion in Section III, and summarize our computations in the algorithm that we present in Section IV. The dynamics equation is then used in the control scheme of Section V, and Section VI demonstrates some example humanoid motions synthesized through this scheme. We conclude the paper in Section VII. In the Appendix Section, a synthetic view of all the notations of the paper is provided.

II. PRELIMINARY COMPUTATIONS

Let us consider an articulated mechanism made of \( n + 1 \) rigid bodies (links) indexed by the variable \( k \) in \( \{0, \ldots, n\} \), the body \( k = 0 \) being the free-floating base link of the mechanism, and that are articulated through \( m \) revolute or prismatic joints indexed by the variable \( j \) in \( \{1, \ldots, m\} \). The general contact state of the mechanism is modeled as follows. On each body \( k \in \{0, \ldots, n\} \) a set of \( n_k \) contact forces \( f_{k,1}, \ldots, f_{k,n_k} \) are applied at the respective body-frame-expressed\(^1\) contact points \( a_{k,1}, \ldots, a_{k,n_k} \) belonging to the surface cover of the body \((n_k = 0 \text{ if there is no contact on body } k)\). We emphasize here the fact that, in the case of an anthropomorphic system, these contact forces are not restricted to be applied on the foot bodies.

We then consider the following generalized coordinates of the system (for convenience, we will use throughout the paper the same notations as in [1] when applicable). The base link of the system, \( k = 0 \), is parameterized with a global position variable \( x_0 \in \mathbb{R}^3 \) and a global orientation variable \( \theta_0 \in \mathbb{R}^d \), \( d \) being the dimensionality of the parameterization we choose for the \( SO(3) \) manifold, see Table I.

<table>
<thead>
<tr>
<th>( \theta_0 )</th>
<th>( d = \dim(\theta_0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler angles (roll-pitch-yaw)</td>
<td>( d = 3 )</td>
</tr>
<tr>
<td>Unit quaternion (Euler parameters)</td>
<td>( d = 4 )</td>
</tr>
<tr>
<td>Rotation matrix</td>
<td>( d = 9 )</td>
</tr>
</tbody>
</table>

Finally the rest of the articulated configuration of the mechanism forms the internal joint angle vector \( \hat{q} \in \mathbb{R}^m \). All in all, the generalized coordinates of the system are expressed in the concatenated vector

\[
q = (x_0, \theta_0, \hat{q}) \in \mathbb{R}^{3+d+m} \tag{1}
\]

Let \( R_0 \) denote the orientation matrix of the body \( k \), \( \omega^\text{body}_0 \) be its body-frame-expressed angular velocity with respect to the inertial frame, and \( J_k \) be the inertial-frame-expressed global (i.e. relative to the inertial frame) position of the origin of the body frame. In particular, \( R_0 \) denotes the orientation matrix of the base link. Let \( \omega^\text{global}_0 \) and \( \omega^\text{base}_0 \) denote, respectively, the global-frame-expressed and base-link-frame-expressed angular velocity vector of the base link body with respect to the inertial (global) frame. These vectors are related to the time derivative of \( \theta_0 \) through the two following linear mappings

\[
\omega^\text{global}_0 = J^\text{global}_0 \dot{\theta}_0 \quad \text{and} \quad \omega^\text{base}_0 = J^\text{base}_0 \dot{\theta}_0 \tag{2}
\]

We now propose a calculation method for the two matrices \( J^\text{global}_0 \) and \( J^\text{base}_0 \), for any arbitrarily chosen parameterization \( \theta_0 \). The angular velocity vector \( \omega^\text{global}_0 \) can be defined as the vector \( \omega^\text{global}_0 \in \mathbb{R}^3 \) such that

\[
[\omega^\text{global}_0 \times] = R_0 R_0^T ,
\]

where, for a vector \( u \in \mathbb{R}^3 \), we use the notation

\[
[u \times] = 
\begin{pmatrix}
0 & -u_3 & u_2 \\
 u_3 & 0 & -u_1 \\
 -u_2 & u_1 & 0 \\
\end{pmatrix} .
\tag{4}
\]

Moreover, we can show that the following fundamental relation holds, \( \forall u \in \mathbb{R}^3 \):

\[
\frac{\partial (R_0 u)}{\partial \theta_0} = -[\mathbf{R}(R_0 u) \times] J^\text{global}_0 = -[R_0 u \times] R_0^T J^\text{global}_0 .
\tag{5}
\]

By applying eq.(5) on the canonical basis vectors of \( \mathbb{R}^3 \), we get, denoting \( R_{0,i} \) the \( i^\text{th} \) column of \( R_0 \) for \( i \in \{1, 2, 3\} \),

\[
\frac{\partial R_{0,i}}{\partial \theta_0} = -[R_{0,i} \times] J^\text{global}_0 .
\tag{6}
\]

Let now \((e_1, e_2, e_3)\) be any orthonormal basis of \( \mathbb{R}^3 \), from the properties of the scalar triple product it follows that, for any vector \( u \in \mathbb{R}^3 \):

\[
u = (u \times e_1) e_3 + (u \times e_2) e_1 + (u \times e_3) e_2 .
\tag{7}
\]

This identity is extendible to any \( 3 \times 3 \) matrix \( M \),

\[
M = -e_3 e_2^T [e_1 \times] M - e_2 e_1^T [e_3 \times] M - e_1 e_3^T [e_2 \times] M .
\tag{8}
\]

\( R_0 \) being an orthogonal matrix, its columns form an orthonormal basis of \( \mathbb{R}^3 \), thus we can apply eq.(8) to get

\[
J^\text{global}_0 = -R_{0,3} R_{0,2}^T [R_{0,1} \times] J^\text{global}_0 - R_{0,2} R_{0,1}^T [R_{0,3} \times] J^\text{global}_0
\tag{9}
\]

\[
- R_{0,3} R_{0,2}^T [R_{0,1}^T \times] J^\text{global}_0.
\]

Finally, replacing the corresponding expressions from eq.(6), we get our final expression

\[
J^\text{global}_0 = R_{0,3} R_{0,2}^T \frac{\partial R_{0,1}}{\partial \theta_0} + R_{0,2} R_{0,1}^T \frac{\partial R_{0,3}}{\partial \theta_0} + R_{0,1} R_{0,3}^T \frac{\partial R_{0,2}}{\partial \theta_0} .
\tag{10}
\]

Note that the latter expression requires the computation of the derivatives \( \frac{\partial R_0}{\partial \theta_0} \), which can be easily obtained from the mapping \( \rho : \theta_0 \to R_0 \) that can be found in introductory kinematics textbooks (writing a rotation matrix as a function of the rotation parameterization). For instance, let us consider
a unit quaternion parameterization, \( \theta_0 = (\alpha, \beta, \gamma, \delta) \), then the expression of the mapping \( \rho \) is given as \( R_0 = \rho(\theta_0) = 
abla \begin{pmatrix} 2(\alpha^2 + \beta^2) - 1 & 2(\beta \gamma - \alpha \delta) & 2(\beta \delta + \alpha \gamma) \\ 2(\beta \gamma + \alpha \delta) & 2(\alpha^2 + \gamma^2) - 1 & 2(\gamma \delta - \alpha \beta) \\ 2(\beta \delta - \alpha \gamma) & 2(\gamma \delta + \alpha \beta) & 2(\alpha^2 + \delta^2) - 1 \end{pmatrix} \) (11)

And the \( \partial R_k / \partial \theta_0 \) can be derived as

\[
\begin{align*}
\frac{\partial R_{i,j}}{\partial \theta_0} &= \begin{pmatrix}
4\alpha & 4\beta & 0 & 0 \\
2\alpha & 2\beta & 2\alpha & 2\alpha \\
-2\gamma & -2\delta & -2\alpha & 2\beta \\
2\beta & 2\gamma & 2\delta & 2\gamma \\
4\alpha & 4\beta & 0 & 0 \\
-2\delta & 2\beta & 2\alpha & 2\beta \\
2\beta & 2\gamma & 2\delta & 2\gamma \\
-2\beta & -2\gamma & 0 & 4\delta
\end{pmatrix},
\end{align*}
\]

Now that we have computed \( J_{\text{base}}^{\text{total}} \) in eq. (10), we have

\[
J_{\omega k}^{\text{base}} = R_0^T J_{\omega k}^{\text{total}}. \quad (13)
\]

Back to our free-floating-base mechanism. For a body \( k \in \{0, \ldots, n\} \) and a point belonging the bodies with coordinates \( p \in \mathbb{R}^3 \) expressed in the body frame, we denote \( J_k(p) \), \( J_{k}^{\text{base}} \), \( J_{\text{local}}^{\text{base}} \), respectively, the translational Jacobian matrix at \( p \), the body-frame-expressed rotational Jacobian matrix, and the inertial-frame-expressed rotational Jacobian matrix, of the body \( k \) relative to the inertial frame with respect to \( q \). Similarly, \( J_k(p) \), \( J_{k}^{\text{base}} \), \( J_{\text{local}}^{\text{base}} \), denote, respectively, the translational Jacobian matrix at \( p \), the base-link-frame-expressed rotational Jacobian matrix, and the body-frame-expressed rotational Jacobian matrix, of the body \( k \) relative to the base link body with respect to \( q \) (in other words, hatted Jacobian matrices are for the articulated motion with respect to the base link, non-hatted Jacobian matrices are for the free-floating-base articulated motion with respect to the inertial frame, thus the latter taking into account the contribution of the base-link motion). Algorithms for the computation of the matrices \( J_k(p) \), \( J_{k}^{\text{base}} \), \( J_{\text{local}}^{\text{base}} \), also known as geometric or kinematic Jacobian matrices, are widely available in the kinematics literature [11] [12]. We show in the sequel how these matrices can be used to compute the less common matrices \( J_k(p) \), \( J_{k}^{\text{base}} \), \( J_{\text{local}}^{\text{base}} \). First we have, for the translational Jacobian matrix

\[
J_k(p) = \begin{bmatrix} 1_{3 \times 3} & - (x_k + R_k p - x_0) \times R_0 J_{\omega 0}^{\text{base}} & R_0 J_{\dot{k}}(p) \end{bmatrix} \quad (14)
\]

Remark 1: If we denote \( p^0 = R_0^T (x_k + R_k p - x_0) \) as the base-link-frame-expressed position of \( p \), which does not depend on \( \theta_0 \), then we can show that

\[
- (x_k + R_k p - x_0) \times R_0 J_{\omega 0}^{\text{base}} = - [R_0^{p^0} \times] J_{\omega 0}^{\text{local}} = [\partial R_{i,j} / \partial \theta_0] p^0 \]

This gives as an alternative scheme for computing the \( d \) middle columns in eq.(14).

We also have, for the rotational Jacobian matrices,

\[
J_{\omega k}^{\text{base}} = [0_{3 \times 3} | R_k^T R_0 J_{\omega 0}^{\text{base}} | \dot{J}_{\omega k}^{\text{base}}] \quad (16)
\]

\[
J_{\omega k}^{\text{base}} = [0_{3 \times 3} | J_{\omega 0}^{\text{base}} | R_0 J_{\dot{k}}^{\text{base}}] \quad (17)
\]

Remark 2: Expression (17) is consistent with the composition rule of angular velocities

\[
\omega_k^{\text{local}} = J_{\omega k}^{\text{base}} \omega, \quad \omega_k^{\text{global}} = J_{\omega k}^{\text{base}} \omega + R_0 \dot{J}_{\omega k}^{\text{base}}, \quad \omega_k^{\text{global}} = \omega_k^{\text{base}},
\]

where \( \omega_k^{\text{base}} \) is the base-link-frame-expressed rotational velocity of the body \( k \) with respect to the base link.

Remark 3: We can easily verify that the two following transport formulas are consistent with the above derivations of the translational and rotational Jacobian matrices:

\[
J_k(a) = J_k(b) - R_k[(a - b) \times] J_{\omega k}^{\text{base}} \quad (18)
\]

\[
\dot{J}_k(a) = \dot{J}_k(b) - R_k^T R_k[(a - b) \times] J_{\omega k}^{\text{base}} \quad (19)
\]

To conclude this section, we denote \( c_k \) the body-frame-expressed center of mass of the body \( k \), \( m_k \) its mass, and \( I_k(p) \) its inertia matrix expressed at the body-frame-expressed point \( p \). We recall that

\[
I_k(c_k) = I_k(0) - m_k (c_k C c_k) 1_{3 \times 3} - c_k \otimes c_k \quad (20)
\]

Let \( g \) denote the gravity vector, and \( u \in \mathbb{R}^m \) denote our torque control input acting on the actuated joints of the mechanism (i.e. internal joints, \( \dim(u) = \dim(\dot{q}) = m \)).

III. GAUSS’ PRINCIPLE WITH MANIFOLD EQUALITY CONSTRAINTS

Following [1], we apply Gauss’ principle of least constraint [13] (also used in robotics for redundant robots in [14], or for simulation in [15]) which states that at a given state of the system (given position and velocity), the acceleration of a constrained articulated system deviates the least from the acceleration it would have in the absence of any constraints induced by the presence of the joints linking the bodies. The deviation is measured through the following quantity

\[
D = \sum_{k=0}^{n} \frac{1}{2} (\ddot{x}_k - \ddot{\tilde{x}}_k)^T m_k (\ddot{x}_k - \ddot{\tilde{x}}_k) + \sum_{k=0}^{n} \frac{1}{2} (\ddot{\omega}_k - \ddot{\tilde{\omega}}_k)^T I_k(c_k) (\ddot{\omega}_k - \ddot{\tilde{\omega}}_k),
\]

where the underlined quantities are the accelerations of the bodies if they were not linked together.
Without the equality constraints that stem from the parameterization of the orientation of the base link \( \theta_0 \), this principle would translate into the following optimality condition

\[
\frac{\partial D}{\partial \dot{q}} = 0, 
\]  
(23)

which can be rewritten in the form (see [1] for details of the derivation 2),

\[
M(q) \ddot{q} + N(q, \dot{q}) \dot{q} = M(q) \begin{pmatrix} g \\ 0_d \\ 0_m \end{pmatrix} + \begin{pmatrix} 0_3 \\ 0_d \end{pmatrix} u + \sum_{k=0}^{n} \sum_{i=1}^{n_k} J_{ik}(a_{k,i})^T f_{k,i}, 
\]  
(24)

where the expressions of the matrices \( M \) and \( N \) are given as

\[
M(q) = \sum_{k=0}^{n} J_{ik}(c_k)^T m_k J_{ik}(c_k) + J_{ik}^{\text{rot}} I_k(c_k) J_{ik}^{\text{rot}}, 
\]  
(25)

\[
N(q, \dot{q}) = \sum_{k=0}^{n} J_{ik}(c_k)^T m_k J_{ik}(c_k) + J_{ik}^{\text{rot}} I_k(c_k) J_{ik}^{\text{rot}} - J_{ik}^{\text{rot}}(I_k(c_k) J_{ik}^{\text{rot}} \dot{q}) \times J_{ik}^{\text{rot}}. 
\]  
(26)

However, the choice of a particular manifold (algebraic variety) parameterization of the rotation group \( SO(3) \) brings along a set of equality constraints

\[
C(q) = 0, 
\]  
(27)

that act on the set of parameters \( \theta_0 \), \( C : \mathbb{R}^{3+d+m} \rightarrow \mathbb{R}^{d-\text{dim}(SO(3))} \) being a mapping of dimension \( \text{dim}(C) = d - \text{dim}(SO(3)) = d - 3 \), see Table II (note: in the last row we did not remove redundant (symmetric) constraints for the sake of notation, we also do not take into account \( \det(\theta_0) = +1 \) that the initial state satisfies this latter condition and that we stay throughout the continuous motion in the same connected component of \( O(3) \)).

<table>
<thead>
<tr>
<th>( \theta_0 )</th>
<th>( \text{dim}(C) )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler angles (roll-pitch-yaw)</td>
<td>0</td>
<td>( C(q) = \theta_0^2 \theta_0 - 1 )</td>
</tr>
<tr>
<td>Unit quat. (Euler parameters)</td>
<td>1</td>
<td>( C(q) = \theta_0^2 \theta_0 - 1 )</td>
</tr>
<tr>
<td>Rotation matrix</td>
<td>6</td>
<td>( C(q) = \theta_0^2 \theta_0 - 1_{3 \times 3} )</td>
</tr>
</tbody>
</table>

For instance, let us consider the case of the unit quaternion parameterization. In that case \( C \) takes the form

\[
C(q) = q^T S^T S q - 1, 
\]  
(28)

where \( S \) is a selection matrix that selects the \( \theta_0 \) component in \( q \)

\[
S = \begin{pmatrix} 0_{d \times 3} & 1_{d \times d} & 0_{d \times m} \end{pmatrix}. 
\]  
(29)

\(^2\)noting that \( m_k \Sigma_k = \sum_{i=0}^{n_k} f_{k,i} \times \dot{x}_k \times (I_k \omega_k^{b_{ih}} - (J_k \omega_k^{b_{ih}}) \times \omega_k^{b_{ih}}) = \sum_{i=0}^{n_k} a_{k,i} \times f_{k,i} \times J_k(c_k) \dot{q} + J_k(c_k) \dot{q} \times J_k(c_k) \dot{q} \).

We then time-differentiate eq.(27) twice to get a constraint on the variable \( \ddot{q} \)

\[
\dddot{C}(\ddot{q}) = q^T S^T S \ddot{q} + q^T S^T \dot{q} = 0. 
\]  
(30)

The optimality condition eq.(23) becomes now, taking into account the constraint eq.(30),

\[
\begin{cases}
\frac{\partial D}{\partial \ddot{q}} - \frac{\partial \ddot{C}}{\partial \ddot{q}} \lambda = 0, \\
\dddot{C}(\ddot{q}) = 0,
\end{cases} 
\]  
(31)

where \( \lambda \in \mathbb{R}^{d-3} \) is the Lagrange multiplier associated with the constraint eq.(30). The equation of motion eq.(24) thus becomes:

\[
M(q) \ddot{q} + N(q, \dot{q}) \dot{q} = M(q) \begin{pmatrix} g \\ 0_d \\ 0_m \end{pmatrix} + \begin{pmatrix} 0_3 \\ 0_d \end{pmatrix} u + \sum_{k=0}^{n} \sum_{i=1}^{n_k} J_{ik}(a_{k,i})^T f_{k,i} + S^T S q \lambda, 
\]  
(32)

where the matrices \( M \) and \( N \) keep the exact same forms as in eq.(25) and eq.(26) respectively, and where we can solve for the Lagrange multiplier \( \lambda \) by pre-multiplying eq.(32) by \( q^T \) as follows [16]:

\[
\lambda = q^T \left[ M(q) \ddot{q} + N(q, \dot{q}) \dot{q} - M(q) \begin{pmatrix} g \\ 0_d \\ 0_m \end{pmatrix} + \begin{pmatrix} 0_3 \\ 0_d \end{pmatrix} u - \sum_{k=0}^{n} \sum_{i=1}^{n_k} J_{ik}(a_{k,i})^T f_{k,i} \right]. 
\]  
(33)

Let us define the following matrix

\[
A = 1_{(3+d+m) \times (3+d+m)} - S^T S q q^T. 
\]  
(34)

By re-injecting eq.(33) into eq.(32) we get our final dynamics equation of motion expression

\[
A [M(q) \ddot{q} + N(q, \dot{q}) \dot{q}] = A \begin{pmatrix} g \\ 0_d \end{pmatrix} + \begin{pmatrix} 0_3 \\ 0_d \end{pmatrix} u + \sum_{k=0}^{n} \sum_{i=1}^{n_k} J_{ik}(a_{k,i})^T f_{k,i}. 
\]  
(35)

It is worth pointing out that in eq.(35), \( \ddot{q} = (\ddot{\theta}_0, \ddot{\theta}_0, \ddot{\theta}_0) \) is the actual second time-derivative of \( q \). In particular, \( \theta_0 \) is simply obtained by time-differentiating twice the parameters in \( \theta_0 \).

**IV. DYNAMICS ALGORITHM**

The proposed algorithm here consists simply in carrying out the computations presented in the two previous sections in the order outlined in the following algorithm:

1. Compute eq.(10) then eq.(13)
Remark 4: Without need of analytically solving for the equivalent quadratic program as follows:

\[ \min_{\tilde{q}, u, f} \sum_i \left| \tilde{r}_i - \tilde{r}_i^{\text{desired}} \right|^2 \]

subject to

\[ u \in U \]
\[ f \in K \]
\[ \forall k, i : J_{tk}(a_k,i)^{\bar{q}} + \dot{J}_{tk}(a_k,i)^{\bar{q}} = 0 \]
\[ \text{Eq. (35)} \]

where \( r_i \) are a set of control tasks (position of a point of one of the bodies, position of the center of mass of the robot, whole configuration of the robot, etc.), \( U \) are the bounds on the actuation torques, and \( K \) are the linearized Coulomb friction cones at the contact points \( a \).

Taking now into account the new equation of motion eq.(35) along with the constraint eq.(30) that happens to be a linear constraint in \( \bar{q} \), we still have a quadratic program formulation, and eq.(36) becomes now

\[ \min_{\tilde{q}, u, f} \sum_i \left| \tilde{r}_i - \tilde{r}_i^{\text{desired}} \right|^2 \]

subject to

\[ u \in U \]
\[ f \in K \]
\[ \forall k, i : J_{tk}(a_k,i)^{\bar{q}} + \dot{J}_{tk}(a_k,i)^{\bar{q}} = 0 \]
\[ \text{Eq. (35)} \]
\[ q^T S^T S \bar{q} + q^T S^T S \bar{q} = 0. \]

Remark 4: Without need of analytically solving for the Lagrange multiplier in eq.(32), we can still formulate an equivalent quadratic program as follows:

\[ \min_{\tilde{q}, u, f, \lambda} \sum_i \left| \tilde{r}_i - \tilde{r}_i^{\text{desired}} \right|^2 \]

subject to

\[ u \in U \]
\[ f \in K \]
\[ \forall k, i : J_{tk}(a_k,i)^{\bar{q}} + \dot{J}_{tk}(a_k,i)^{\bar{q}} = 0 \]
\[ \text{Eq. (32)} \]
\[ q^T S^T S \bar{q} + q^T S^T S \bar{q} = 0. \]

V. APPLICATION TO WHOLE-BODY CONTROL OF HUMANOID ROBOT

In [17] we presented in simulation a control scheme for free-floating-base humanoid robot in general contact configuration with the environment by solving at each time step of the simulation the following quadratic program [18] [19].

\[ \min_{\tilde{q}, u, f} \sum_i \left| \tilde{r}_i - \tilde{r}_i^{\text{desired}} \right|^2 \]

subject to

\[ u \in U \]
\[ f \in K \]
\[ \forall k, i : J_{tk}(a_k,i)^{\bar{q}} + \dot{J}_{tk}(a_k,i)^{\bar{q}} = 0 \]
\[ \text{Eq. (24)} \]

where \( r_i \) are a set of control tasks (position of a point of one of the bodies, position of the center of mass of the robot, whole configuration of the robot, etc.), \( U \) are the bounds on the actuation torques, and \( K \) are the linearized Coulomb friction cones at the contact points \( a \).

As an application, we incorporated this equation, together with the manifold equality constraints, as linear constraints in a quadratic program formulation used for the control and simulation of humanoid robot in various contact configurations between the robot links and the environment. Example motions on single and double-support stances and on handstand were achieved.

The same framework can be used to handle spherical joints (ball-and-socket joint) that are commonly used in the modeling of virtual human characters in computer graphics, prototyping simulation, and biomechanics modeling applications. Future work can investigate this possibility.

VI. EXAMPLES

We improved the motions demonstrated in [17] by applying the new control scheme eq.(37) to the humanoid robot HRP-2 [20] with unit-quaternion-parameterized waist link (the base link) in simulation. Motions on different contact configurations (feet/hands) were dynamically simulated without needing to re-parameterize the configuration of the robot depending on the current contact state. See the attached video and Figs. 2, 3, and 4.

In this video, the sequence of static postures used as input and displayed before each motion is autonomously planned offline using the multi-contact stance planning algorithm in [21] after the user have specified an initial and goal posture or initial and goal set of contact locations. The tasks \( r_i \) are decided online by a finite-state machine as described in [17], these tasks track the position of the CoM and the overall posture of the robot in the next static posture whenever a static posture of the sequence is reached, and an additional task tracks the position of the limb being moved to a new contact location.

As for computation time, the problem eq.(37) is solved at every simulation step, of 1ms, in an average of approximately 7ms, on a 3.06GHz Intel Pentium IV system with 1Gb of RAM memory.

VII. CONCLUSION

We presented a simple algorithm for the computation of the dynamics equation of motion of free-floating-base mechanisms using arbitrary manifold parameterization of the \( SO(3) \) group in which the free-floating base of the mechanism evolves rotationally. We instantiated our derivations for the example singularity-free low-dimensional unit-quaternion parameterization of this rotation manifold.

The derivation of the equation of motion followed Gauss’ principle of least constraint, taking into account, when writing the optimality condition for this principle, the additional equality constraints acting on the parameters of the chosen manifold coordinate chart. These equality constraints resulted in the emergence of Lagrange multipliers that were solved for and eliminated from the new equation of motion.

As an application, we incorporated this equation, together with the manifold equality constraints, as linear constraints in a quadratic program formulation used for the control and simulation of humanoid robot in various contact configurations between the robot links and the environment. Example motions on single and double-support stances and on handstand were achieved.

The same framework can be used to handle spherical joints (ball-and-socket joint) that are commonly used in the modeling of virtual human characters in computer graphics, prototyping simulation, and biomechanics modeling applications. Future work can investigate this possibility.

Finally, another promising direction for further research building on these results is the modeling of manipulation systems with free-floating manipulated objects using the
same common formulation for the dynamics of the system made of the manipulator and the object.

**APPENDIX: NOTATION TABLE**

| $n + 1$ | Number of bodies |
| $k = 0$ | Base link |
| $m$ | Number of Joints |
| $q$ | Configuration vector |
| $\dot{q}$ | Internal (actuated) joint vector |
| $\theta_0$ | Orientation of the base link |
| $d$ | $\dim(\theta_0)$ |
| $x_k$ | Position of the body $k$ |
| $\omega_k$ | Angular velocity of the body $k$ |
| $R_k$ | Orientation matrix of the body $k$ |
| $\Theta_{local}$ | expressed in the body frame |
| $\Theta_{base}$ | expressed in the base-link frame |
| $\Theta_{global}$ | expressed in the inertial (global) frame |
| $J_{\omega_0}$ | Mapping from $\theta_0$ to $\omega_0$ |
| $\rho$ | Mapping from $\theta_0$ to $R_0$ |
| $J_{\delta k}$ | Trans. Jac. Mat. of the body $k$ with resp. to $q$ |
| $J_{\delta k}$ | Rot. Jac. Mat. of the body $k$ with resp. to $q$ |
| $J_{\delta k}$ | Trans. Jac. Mat. of the body $k$ with resp. to $\dot{q}$ |
| $J_{\delta k}$ | Rot. Jac. Mat. of the body $k$ with resp. to $\dot{q}$ |
| $f_{k,i}$ | Contact forces applied on the body $k$ |
| $\omega_{k,i}$ | Application point of the contact force $f_{k,i}$ |
| $\delta_k$ | Deviation from the unconstrained motion |
| $M, N$ | Dynamics quantities |
| $C$ | Constraint on $q$ (on $\theta_0$) |
| $\lambda$ | Lagrange multiplier |
| $S$ | Selection matrix (of the $\theta_0$ component in $q$) |
| $u$ | Actuation torques (control input) |
| $g$ | Gravity vector |
| $\kappa$ | local-frame expressed CoM of the body $k$ |
| $m_k$ | Mass of the body $k$ |
| $I_k$ | Inertia matrix of the body $k$ |
| $\tau$ | Control task (objective, feature) |
| $U$ | Torque limits |
| $K$ | Linearized Coulomb friction cone |
| $A$ | Matrix accounting for the equality constraint |
| $0$ | Zero vector or matrix |
| $1$ | Identity matrix |

**ACKNOWLEDGMENT**

This work is partially supported by grants from the RoboHow.Cog FP7 www.robohow.eu, and by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (B), 22300071, 2010. The first author is supported with a JSPS Postdoctoral Fellowship for Foreign Researchers, ID No. P12707.

**REFERENCES**


Fig. 2. Motion on single-support and double-support stances on feet.

Fig. 3. Motion on single-support and double-support stances on hands.

Fig. 4. Motion on single-support and double-support stances on feet.