On Humanoid Motion Optimization
Wael Suleiman, Eiichi Yoshida, Jean-Paul Laumond, André Monin

To cite this version:

HAL Id: hal-00164004
https://hal.archives-ouvertes.fr/hal-00164004v3
Submitted on 29 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
On Humanoid Motion Optimization

Wael Suleiman†, Eiichi Yoshida‡, Jean-Paul Laumond* and André Monin*
IS/AIST-ST2I/CNRS Joint Japanese-French Robotics Laboratory (JRL)
† LAAS - CNRS, university of Toulouse
7 Avenue du Colonel Roche, 31077 Toulouse, France
{suleiman, jpl, monin}@laas.fr
‡ Intelligent Systems Research Institute, National Institute
of Advanced Industrial Science and Technology (AIST)
AIST Central 2, Umezono 1-1-1, Tsukuba, 305-8568 Japan
{e.yoshida}@aist.go.jp

Abstract—In this paper, we present a recursive method for the
optimization of humanoid robot motions. The method is based
on an efficient dynamics algorithm, which allows the calculation
of the gradient function with respect to the control parameters
analytically. The algorithm makes use of the theory of Lie
groups and Lie algebra. The main objective of this method is
to smooth the pre-calculated humanoid motions by minimizing
the efforts, and at the same time improving the stability of
the humanoid robot during the execution of the planned tasks.
Experimental results using HRP-2 platform are provided to
validate the proposed method.

Index Terms—Humanoid robot; Motion planning; Recursive
dynamics; Optimization

I. INTRODUCTION

Few years ago, talking about humanoid robots was some-
kind of science fiction. The recent technological advancement
has made this dream a reality. Actually the ability of humanoid
robots to execute complex tasks increases rapidly.

The latest trends in humanoid research are to increase their
autonomous behavior as well as improving the stability and
smoothness of the planned motions.

Optimizing motions to improve their performance was an
active research subject in recent years. In virtual reality, Lo
and Metaxas [1] have proposed a method based on optimal
control theory within a recursive dynamics framework. The
objective of their work is to simulate a dynamically-correct
astronaut motions by minimizing joint torques.

In humanoid research, Ruchanurucks et al [2] have proposed
a method to optimize upper body motion of humanoid robot
in order to imitate a recorded human motion. Their optimization
objective function preserves the salient characteristics of the
original motion, and at the same time it respects the physical
constraints of the humanoid robot. However, the notion of
minimizing an energy function is not introduced because the
objective is only to guarantee the feasibility of the motions.
Furthermore, the authors have mentioned that the resulting
trajectories would meet the latter limits while the former limits
are often violated. This is because their method consider the
velocity and force constraints separately.

Safronova et al [3] use also a pre-recorded human motion to
generate optimal motion of the upper body of Sarcos humanoid
robot. The objective is also to imitate human motions, so
the function to be minimized is the difference between the
recorded and the executed motion by the robot. Guilamo et
The optimal solution is in kinematic sense and does not take
into account the dynamic constraints.

Sentsis and Khatib [5] have proposed a whole-body control
framework for humanoids. This framework integrates task-
oriented dynamic control while complying with humanoid
physical constraints. The controllers are calculated in the
operational space [6] at multiple levels. As it is known, these
controllers provide the torques which should be applied on
each joint, that means the humanoid robot should be controlled
by joint torques. On the contrary, HRP-2 platform and many
humanoid robots are controlled by joint positions. Although
the use of inverse dynamics methods can give the associated
joint positions with the calculated joint torques, this method
is time consuming and is not numerically efficient.

In this paper, our objective is to smooth pre-calculated hu-
manoid motions. These motions can be provided by kinematic
planning methods, which can take into account the limits of
joint angles. On the other hand, they can not guarantee that
the calculated motions do not violate the torques limits.

Studying human movements [7], [8] has brought out a
connection between minimizing energy dissipation and forces,
and the smoothness of human movements. On account of the
complexity of calculating the energy dissipation, one can use
an approximative prediction of it. A good predictor of human’s
metabolic energy is proven to be the joint torques [9]. As the
humanoid robot is supposed to realize human-like motions,
our goal can be achieved by minimizing the joints torques
during the planned motion.

The contributions of this paper are:

• Developing an optimization framework for humanoid
robot motions. This framework takes as input a pre-
calculated motions, which are provided by motion planning
techniques. The output is an optimized and stable
motions.

• The proposed method uses the forward dynamic formulation
and the quantities to be optimized are the joint
positions. As a consequence, the humanoid robot can be
controlled directly in the joint space and not in torque
control space. Therefore the method is well adapted for a position controlled humanoid like HRP-2 platform.

- By using an efficient dynamic algorithm, we can calculate the derivative of joint torques with respect to joint position, velocity and acceleration \((q, \dot{q}, \ddot{q})\) analytically. This procedure is similar to the procedure used in [1]. However, the problem of virtual human stability and ground reaction forces modeling are not considered in [1].
- The validation of the proposed method pointed out that the optimized motion is smoother than the pre-calculated one.

The remainder of this paper is organized as follows. In Section II the kinematic structure of HRP-2 humanoid robot, and the definition of active and passive parts are introduced. An overview of the algorithm of recursive multibody dynamics is given in Section III. In Section IV the optimization problem is formulated. In Section V the discretization of configuration space and solving the optimization problem are pointed out. Experimental results are given in Section VI and Section VII concludes the paper.

II. HUMANOID ROBOT: KINEMATIC STRUCTURE

The kinematic structure of the humanoid robot HRP-2 [10] is given in the Fig. 1. In this structure the degree of freedoms are presented by cylinders. The structure contains 30 degree of freedoms.

![Fig. 1. Description of HRP2 kinematic structure.](image)

Our objective is to optimize pre-calculated humanoid motions and improve their stability. The pre-calculated motions are the output of motion planning algorithms [11], [12]. These motions are generally complex, e.g. transporting an object and guaranteeing the collision avoidance [13]. Such motions usually use the upper part of humanoid robot to execute the desired task and the lower part is mainly responsible of the locomotion and maintaining the stability of humanoid robot.

For that, we divide the kinematic structure into two parts. The first part is the active part, which is the lower part. The upper body is regarded as passive part.

In order to achieve our objective, we optimize only the values of the degree of freedoms of the active part. The passive part will be taken into account in the calculation of the dynamic equations without modifying their angular values. The active part consists of 12 degree of freedoms.

III. RECURSIVE MULTIBODY DYNAMICS

Park et al [14] have proposed to write the recursive multibody dynamics for serial open or branched kinematic chains using Lie group and Lie algebra. The main advantage of this formulation is to relate the joint torques and joint angles explicitly. Therefore the differentiation of joint torques with respect to joint angles can be done analytically.

Let us define the Lie groups \(SO(3)\) and \(SE(3)\), which denote the orthonormal matrix \(\Theta\) in \(R^{3 \times 3}\) and the homogeneous transformation group respectively. The Lie algebra of \(SO(3)\) and \(SE(3)\) are denoted \(so(3)\) and \(se(3)\) respectively. The operators defined on these groups are: skew, matrix exponential, adjoint map \(Ad_{q}()\), dual adjoint \(Ad^*_{q}()\), Lie bracket \(ad_{q}()\) and dual Lie bracket \(ad^*_{q}()\). For more details on Lie group, Lie algebra and the operators definitions see Appendix.

A. Forward Kinematics

The kinematics of an open chain can be modeled as a sequence of homogeneous transformation between consecutive joint frames. Let \(T_{i-1,i} \in SE(3)\) be the transformation matrix between the frame of link \(i\) and the frame of link \(i-1\). The matrix \(T_{i-1,i}\) can be written using matrix exponential notation as follows

\[
T_{i-1,i} = M_i e^{S_q q_i} \tag{1}
\]

where \(S_q \in se(3)\) is the joint screw written in the coordinate of link \(i-1\), \(q_i\) is the current position of joint \(i\) and \(M_i\) is the coordinate transformation between link \(i\) and link \(i-1\).

Using the above definition of transformation matrix, the end-effector of a kinematic chain can be calculated by the product

\[
T_{0,n} = T_{0,1} T_{1,2} \cdots T_{n-1,n} = M_1 e^{S_q q_1} M_2 e^{S_q q_2} \cdots M_n e^{S_q q_n} \tag{2}
\]

Note that by expressing the matrix of transformation in exponential form, we can calculate its derivative with respect to \(q_i\) analytically.

B. Recursive inverse dynamics of branched chains

Branched chains are serial open chains with two or more branches leading to two or more tip links [14], [15]. In the branched chains two definitions arise:

- **Parent link**: the link inward (towards the base) from a given link.
- **Child link**: the link or links which are outward (towards the tips) from a given link.

Spatial velocity of branched chains:

- **Initialization**: Given \(V_0\).
- **Outward recursion**: loop over all links in depth manner:

\[
T_{Pi} = M_{Pi} e^{S_q q_i} \quad V_i = Ad_{T_{Pi}}^{(j)}(V_P) + S_i q_i \quad a_i = -ad_{S_i q_i}(V_i) \quad b_i = -ad_{q_i}^{*}(J_i V_i) \tag{3}
\]
where the index $P$ denotes the parent link of link $i$, $T_{Pj}$ designs the mapping from the link $i$ to its parent $P$ and $V_P$ denotes the spatial velocity of link $P$.

**Applied torques on the branched chains:** In order to calculate the inward recursion of forces and torques, we define the external forces applied on a link $j$ by $F_j$.

- **Initialization:** Given the external applied forces on each link $F_j$, $V_0$ and $J_0 = 0$ for each tip link.
- **Inward recursion:** loop over all links in reversed breadth

$$
\begin{align*}
\dot{V}_i &= A_d T_{Pj}^{-1}(V_P) + S_i \dot{q}_i + a_i \\
\dot{J}_i &= J_i + \sum_{j \in C} A_d T_{i,j}^{-1} J_j A_d T_{i,j}^{-1} \\
B_i &= b_i + \sum_{j \in C} A_d T_{i,j}^{-1} z_j \\
z_i &= \dot{J}_i (S_i \dot{q}_i + a_i) + B_i + \sum_{j \in C} A_d T_{i,j}^{-1} F_j \\
F_i &= J_i A_d T_{Pj}^{-1} (V_P) + z_i \\
\tau_i &= S_i^T F_i
\end{align*}
$$

(4)

where $C$ denotes the child links for link $i$, and $\tau_i$ is the torque applied on the joint $i$.

The matrix $J_i$ is called the spatial inertia and it is defined as follows

$$J_i = \begin{bmatrix} I_i - m_i [r_i]^2 & m_i [r_i] \\ -m_i [r_i] & m_i 1 \end{bmatrix}
$$

(5)

where $I_i$ is the inertia of the link $i$ about its centre of mass and $m_i$ is its mass. $r_i$ is the vector from the point of application of the force and the centre of mass of the link $i$. Recall that $[r_i]$ is the skew operator see Appendix for more details.

**C. Ground reaction forces**

In the absence of external forces applied on the robot, the only forces are the ground reactions forces. In order to estimate the ground reaction forces, we distinguish two cases

1) Single support: The ground reaction force is applied on the support foot and its magnitude is equal to $(Mg - a_G)$, where $M$ is the total mass of the humanoid robot, $g$ is the gravitational acceleration, and $a_G$ is the acceleration of the humanoid robot centre of mass.

2) Double support: The ground reaction forces are applied on the two feet. Their magnitudes are proportional to the distance from the projection of the center of mass on the floor and the centers of the support feet.

A demonstration of the ground reaction forces in single and double support is shown in Fig. 2.

**IV. Optimization problem formulation**

The cost function to be minimized is the integral of the Euclidean norm of joint torques. The stability of the humanoid robot can be assured by guaranteeing that the foot, which is in contact with the ground, will be immobile. Such condition can be satisfied by ensuring that the spatial velocity and acceleration of the support foot are null.

Let $Q_t$ be the vector of angular values of joints in the configuration space defined as follows

$$
Q_t = \begin{bmatrix} q_1 \cr q_2 \cr \vdots \cr q_n \cr p_1 \cr p_2 \cr \vdots \cr p_n \cr r \end{bmatrix}
$$

(6)

where $q_t$ and $p_t$ denote the vectors of angular values of active and passive joints respectively.

The optimization problem can be formulated as follows

$$
\begin{align*}
\min_{Q_t, \dot{Q}_t, \ddot{Q}_t} & \int_0^T \tau_t^T \tau_t dt \\
\text{subject to} & \\
& S_t^T F_t = \tau_t \\
& \dot{Q}_t = \dot{Q}_0, \quad \ddot{Q}_t = 0, \quad \dot{\theta}_0 = 0 \\
& q^0 \leq q_t \leq q^+ \\
& \dot{V}_{\text{support foot}} = 0 \\
& \ddot{V}_{\text{support foot}} = 0 \\
& q_{\text{foot}} = q_{\text{foot}}^{\text{ref}}, \quad \dot{q}_{\text{foot}} = \ldots, \dot{q}_{\text{foot}} = \ldots
\end{align*}
$$

(7)

where $\tau_t$, $F_t$, and $S$ are defined as follows

$$
\tau_t = \begin{bmatrix} \tau_{1,t} \\ \tau_{2,t} \\ \vdots \\ \tau_{n,t} \end{bmatrix}, \quad F_t = \begin{bmatrix} F_{1,t} \\ F_{2,t} \\ \vdots \\ F_{n,t} \end{bmatrix}, \quad S = \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ 0 & S_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & S_n \end{bmatrix}
$$

(8)

$\tau_t$, $F_t$, and $S$ are the vectors of the applied torques and forces on the joints of active part respectively. $\dot{\theta}_0$ and $\ddot{\theta}_0$ denote the value of the applied torque and force on the joint $i$ respectively. $x^-$ and $x^+$ denote the minimal and the maximal values of vector $x$ respectively.

$q_{\text{foot}}^{\text{ref}}$ denotes the configuration of the foot in the euclidean space as shown in Fig. 3

$$
q_{\text{foot}}^{\text{ref}} = \begin{bmatrix} \text{Pos}_{\text{foot}} \\ \text{Pos}_{\text{foot}} \end{bmatrix}
$$

(9)
Thus the optimization problem (7) can be transformed to the following classical form

$$\min_{X_t, \lambda} \frac{\partial L(X_t, \lambda)}{dX_t} = \frac{\partial L(X_t)}{dX_t} + (\lambda_H + \sigma H(X_t)) \frac{\partial H(X_t)}{dX_t} + \max \{0, \lambda_H + \sigma G(X_t)\} T \frac{\partial G(X_t)}{dX_t}$$  \tag{14}$$

As $\lambda^*$ is unknown, an update rule is used

$$\lambda^{k+1}_H = \lambda^*_H + \sigma H(X^*_t)$$

$$\lambda^{k+1}_H = \lambda^*_H + \sigma G(X^*_t)$$  \tag{15}$$

where $X^*_k$ is the unconstrained minimum of $L(X_t, \lambda^*)$. Such updating rule will generate a sequence $\lambda^k$ converges to $\lambda^*$ [17]. In practice, a good schedule is to choose a moderate $\sigma^0$, and increase it as follows

$$\sigma^{k+1} = \alpha \sigma^k$$  \tag{16}$$

where $\alpha$ is between 5 and 10. A threshold $\tilde{\sigma}$ is chosen and the update rule of $\sigma$ stops when $\sigma^k$ becomes higher than $\tilde{\sigma}$.

For more details on the algorithm of augmented Lagrange multiplier method see [18], [16], [17].

Approximating the gradient function $\frac{\partial L(X_t, \lambda)}{dX_t}$ by a numerical difference method is usually used in practice. However, this approach not only a time consuming method on account of the evaluation of the gradient calculation, but also may not converge well because of the approximation.

As we have mentioned the main advantage of using the recursive dynamic algorithm explained in Section III-B is calculating the gradient function analytically in a recursive way.
A. Gradient calculation

The objective is to calculate the gradient of the dynamic quantities, such as \( \tau_i, V_i \) and \( \dot{V}_i \).

By considering the vector of parameters \( X_i = [q_i^T \ \dot{q}_i^T \ \ddot{q}_i^T]^T \), let us start by calculating the derivatives of the operators with respect to an element \( x \) of \( X_i \)

\[
\frac{\partial T_{0,n}}{\partial x} = T_{0,l}(S_i \delta_{x,q_i}) T_{l,n} \\
\frac{\partial A_{d_{T_{i-1}}}}{\partial x} = a_{d_{T_{i-1}}}(S_i \delta_{x,q_i}) + A_{d_{T_{i-1}}}(\frac{\partial Y}{\partial x}) \\
\frac{\partial A_{d_{\beta_{i+1}}}}{\partial x} = a_{d_{\beta_{i+1}}}(S_i \delta_{x,q_i}) \left( A_{d_{T_{i-1}}} \right) + \cdots \\
\frac{\partial A_{d_{f_{i+1}}}}{\partial x} = a_{d_{f_{i+1}}}(S_i \delta_{x,q_i}) \left( A_{d_{T_{i-1}}} \right) + \cdots
\]

(17)

where \( \delta_{x_1,x_2} \) is the Kronecker's delta function defined as follows

\[
\delta_{x_1,x_2} = \begin{cases} 
1 & \text{if } x_1 = x_2 \\
0 & \text{otherwise}
\end{cases}
\]

(18)

The calculation of the gradient with respect to \( X_i \) can be done in a recursive way analogously to the recursive dynamic calculation.

Forward recursion:

- Initialization: Given \( \frac{\partial V_i}{\partial X_i} \).
- loop over all links in depth manner:

\[
\frac{\partial V_i}{\partial X_i} = \frac{\partial A_{d_{T_{i-1}}}}{\partial X_i} (V_p) + S_i \frac{\partial \dot{q}_i}{\partial X_i} \\
\frac{\partial a_i}{\partial X_i} = -\frac{\partial d_{\beta_{i+1}}}{\partial X_i} (V_i) \\
\frac{\partial b_i}{\partial X_i} = -\frac{\partial d_{f_{i+1}}}{\partial X_i} (J_i V_i)
\]

(19)

Backward recursion:

- Initialization: Given \( \frac{\partial \dot{V}_i}{\partial X_i} \).
- loop over all links in reversed breadth

where as we mentioned, earlier, \( C \) denotes the child links for link \( i \).

V. DISCRETIZATION OF CONFIGURATION SPACE

It is well known that the space of the admissible solutions of the minimization problem (7) is very large. In order to transform this infinite dimensional space to a finite one, we can use a basis of shape functions. Let us consider a basis of shape functions \( B_i \) that is defined as follows

\[
B_i = [B_i^1 \ B_i^2 \ \cdots \ B_i^n]^T
\]

(21)

where \( B_i^j \) denotes the value of shape function number \( i \) at the instant \( t \), the dimension of \( B_i \) is \( l \) defines the dimension of the basis of shape functions.

The projection of the vector of angular values \( q_i \) into the basis of shape functions \( B_i \) can be given by the following formula

\[
q_i = Q_B B_i
\]

(22)

where \( Q_B \) is a constant matrix.

The derivative \( \dot{q}_i \) and \( \ddot{q}_i \) can be written as follows

\[
\dot{q}_i = Q_B \dot{B}_i \\
\ddot{q}_i = Q_B \ddot{B}_i
\]

(23)

In this case, the derivative with respect to each element \( Q_B (i,j) \) of the matrix \( Q_B \) can be computed using the following formula

\[
\frac{\partial Q_B(i,j)}{\partial X_i} = \frac{\partial Y}{\partial X_i} \times \frac{\partial X_i}{\partial Q_B(i,j)}
\]

(24)
where $e_i \in \mathbb{R}^m$, 

$$ e_i = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^T $$

and $\otimes$ denotes Kronecker's product operator.

By using the discretization of the configuration space, the optimization problem transforms into the problem of finding the optimal matrix $Q_B$, which minimizes the function $\tilde{L}(X, \lambda)$ in Eq. (13).

It remains to define the shape functions $B_i$. In our case, the shape functions should verify the following properties:

1) They are continuous.
2) Their first and second derivatives are continuous.

Therefore, we use the quartic B-spline functions.

VI. EXPERIMENTAL RESULTS

The experimental scenario that we have tested to validate the proposed method is the following:

1) The robot carries a bar with its right hand.
2) The robot starts walking and depressing the vertical position of its pelvis. At the same time, it lifts up the carried bar.

The characteristics of the carried bar are: length= 2 m, weight= 0.7 Kg, cylindrical form with uniform density distribution. The robot grasps the bar at 0.35 m from its end.

Snapshots of the conducted motion$^1$ are presented in Fig. 4. Note that the objective of using the carried bar is to generate a non-symmetry in the kinematic structure, and also to perturb dynamically the motion by the movement of the bar.

To compare the obtained results with the results obtained by the method presented in [19], we use the real quantities measured by the sensors of the humanoid robot HRP-2.

Fig. 5 shows the $x$ coordinate of the Zero Moment Point (ZMP). The ground reaction force applied on the left foot are presented in Fig. 6. Fig. 7 shows the applied torque on the left knee.

From Figures 5, 6 and 7 we conclude that

1) The optimization method smooths the shape of ZMP, and the oscillations have been avoided. Note that the ZMP trajectory of the optimized motion does not follow the designed ZMP trajectory. This is because the optimization method assures the stability of humanoid robot by guaranteeing that the spacial velocity and acceleration of the support foot are null, and it does not consider the trajectory of ZMP.
2) Using the optimization method avoids the surges in the ground reaction forces applied on the foot.
3) Using the optimization method not only minimizes the joint torques, but also it avoids the surges.

$^1$Videos of experimental results are available at http://www.laas.fr/~suleiman/Humanoids07/index.htm

VII. CONCLUSION

In this paper, we have presented an optimization method for humanoid robot motions. The objective of this method is to smooth and improve the stability of the humanoid motions. To achieve this goal, the integral of Euclidean norm of the applied torques on the joints is minimized, and the dynamical stability conditions are transformed into guaranteeing that the spatial velocity and accelerations of the support foot are null.

The experimental results have pointed out that the proposed method not only smooths the motion but also yields a dynamically stable motion. Future work will focus on the improvement of the proposed method to take into account all degree of freedoms of the humanoid robot. In this case deforming the planned trajectory is required, therefore a
function to calculate the distance between the robot and the obstacles should be developed and considered as an additional constraint.

APPENDIX

A Lie group is a differentiable manifold. An example of Lie group is the orthonormal matrix $\Theta$ in $R^{3\times 3}$, which is called $SO(3)$. Note that this group consists of the rotation matrices in Euclidean space. Another example of Lie group is the group of homogeneous transformation which is the special Euclidean group or $SE(3)$. Given a rotation $\Theta \in SO(3)$ and translation $b \in R^3$, the homogeneous matrix is defined as follows

$$G = \begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix}$$

An important concept associated with each Lie group is the notation of Lie algebra. The tangent space at the identity element of a Lie group is called the Lie algebra for that group. The Lie algebra of $SO(3)$ and $SE(3)$ are denoted $so(3)$ and $se(3)$ respectively.

Let us define some notations and operations on Lie groups and Lie algebra:

1) Skew operator:

$$[\omega] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

2) $(\cdot, \cdot)$ operator:

$$(\omega, v) = \begin{bmatrix} \omega \\ v \\ 0 \end{bmatrix}$$

3) Matrix exponential:

$$e^{(\omega,v)} = \exp \begin{bmatrix} \omega \\ v \\ 0 \end{bmatrix} = \begin{bmatrix} \exp(\omega) & Av \\ 0 & 1 \end{bmatrix}$$
where
\[ \exp([\omega]) = I + \frac{\sin \phi}{\phi} [\omega] + \frac{1 - \cos \phi}{\phi^2} [\omega]^2, \phi = \|\omega\| \]
\[ A = I + \frac{1 - \cos \phi}{\phi^2} [\omega] + \frac{\phi - \sin \phi}{\phi^3} [\omega]^2 \] \hspace{1cm} (A-5)

4) Adjoint map on SE(3):
\[ Ad_G(h) : se(3) \rightarrow se(3)^* \]
\[ Ad_G(h) = \begin{bmatrix} \Theta & 0 \\ [b]\Theta & h_0 \end{bmatrix} \] \hspace{1cm} (A-6)
where \( G \in SE(3) \) is defined as in (A-1), and \( h = (h_0, h_v) \in se(3) \).

5) Dual adjoint operator:
\[ Ad_G^*(h^*) : se(3)^* \rightarrow se(3)^{**} \]
\[ Ad_G^*(h^*) = \begin{bmatrix} \Theta^T & \Theta^T [b]^T \\ 0 & \Theta^T \end{bmatrix} \begin{bmatrix} M \\ F \end{bmatrix} \] \hspace{1cm} (A-7)
where \( G \in SE(3) \), and \( h^* = (M, F) \in se(3)^* \).

6) Lie bracket operator:
\[ ad_g(h) = [g, h] = \begin{bmatrix} [g_0] & 0 \\ [g_v] & [g_0] \end{bmatrix} \begin{bmatrix} h_0 \\ h_v \end{bmatrix} \] \hspace{1cm} (A-8)
where \( g, h \in se(3) \), \( h = (h_0, h_v) \) and \( g = (g_0, g_v) \).

7) Dual Lie bracket operator:
\[ ad_g^*(h^*) = [g, h^*] = \begin{bmatrix} [g_0]^T & [g_v]^T \\ 0 & [g_0]^T \end{bmatrix} \begin{bmatrix} M \\ F \end{bmatrix} \] \hspace{1cm} (A-9)
where \( g = (g_0, g_v) \in se(3) \) and \( h^* = (M, F) \in se(3)^* \).

REFERENCES
