SC 651

A Lie Group Formulation of Robot Dynamics

Authors: F.C.Park, J.E.Bobrow, S.R.Ploen

Parth Pundalik Pai

Roll No. 22B3305

Submission date: 25/04/2024

Research Paper Review - Link to the paper

Indian Institute of Technology Bombay

For mistakes, please reach out to me at 22b3305@iitb.ac.in or parthpai07@gmail.com

Contents

1 Abstract in my viewpoint

This paper gives us a geometric treatment of robot dynamics. We specifically confine ourselves to open-chain manipulators and formulate the equations of motion using the concepts of Lie groups. Generally, these open-chain manipulators lead to complicated equations even for small number of linkages. Here the paper attempts to use concepts of Lie groups in every linkage to possibly get an iterative algorithm or explicit equations so that they are computationally more effective and can be easily differentiated and factored. The paper uses both Newton-Euler and Lagrangian formulations to get both the O(n) iterative algorithm and closed-form equation. This can be later used for robot design, calibration, and motion optimization where an analytic gradient is required.

2 Things before the paper

Before this paper, generating a coupled rigid body open-chain manipulator using Newton-Euler or Lagrange formulation would lead to highly complex equations. There have been numerous proposed formulations to reduce complexity by using advanced computing techniques or making more efficient algorithms. However, the complexity of the design, control, and motion planning of robot manipulators was increasing. Hence this shifted the focus towards more systematic and elegant dynamic formulations that made parameters explicit to facilitate and computation of gradients.

Here we use Lie group along with Riemannian geometry to describe the equations of the Open-chain manipulator. This reduces the need for ad-hoc definitions and notational conventions. This will also lead to a connection between geometric ideas and aspects of robotics. There are prior efforts aimed to "geometrize" both Newton-Euler and Lagrangian formulations of robot dynamics. The field was later moving towards a set of dynamic equations that were closed-form, avoiding complex rules and iterations typical in previous approaches, making them more applicable in various robust control schemes and optimization algorithms in robotics.

3 Summary of the Paper

3.1 Geometric Background

3.1.1 SE(3) and SO(3)

$SE(3)$ is *Special Eucledian Lie Group* defined for rigid-bodies. Any general element in it can be thought of as the following matrix.

$$\begin{bmatrix} \boldsymbol{\Theta} & \mathbf{b} \\ 0 & 1 \end{bmatrix}$$

where $\boldsymbol{\Theta} \in \mathrm{SO}(3)$ and $\mathbf{b} \in \mathbb{R}^3$. More conveniently, it can be represented as an ordered pair ($\boldsymbol{\Theta}$, $\mathbf{b}$). Now the Lie algebra of $SE(3)$ is $\mathfrak{se}(3)$. Any general element in it can be represented in the following manner.

$$\begin{bmatrix} [\omega] & v \\ 0 & 1 \end{bmatrix}$$

where $[\omega] =$

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

This can also be represented as an ordered pair (ω, v) where $\omega = (\omega_1,\omega_2,\omega_3)$

Now we have a couple of ways of representing the element of Lie Group using the element of the corresponding Lie algebra. This can be explained using Adjoint Representation as follows. Suppose G is a Lie group and $\mathfrak{g}$ is the corresponding Lie algebra at the identity element, then the Adjoint map is the invertible map defined in the Lie Algebra.

$$Ad_X : \mathfrak{g} \to \mathfrak{g}$$

$$Ad_X = XxX^{-1}$$

For X = (Θ,b) and x = (ω,v), then

$$Ad_X(x) = \begin{bmatrix} \Theta & 0 \\ [b]\Theta & \Theta \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} \tag{1}$$

One can also define an Adjoint map in dual space of lie algebra. For some z = (m, f) we have,

$$Ad^*_X(x) = \begin{bmatrix} \Theta^T & \Theta^T[b]^T \\ 0 & \Theta^T \end{bmatrix} \begin{bmatrix} m \\ f \end{bmatrix} \tag{2}$$

There is another way of representing an element of Lie group, that is by using Lie Brackets. They are as follows. Suppose we have a lie algebra g of a Lie group G, and x $\in$ G, then we define adjoint representation as

$$ad_x(y) = [x, y] = xy - yx$$

Similarly, we can have the dual space version of the adjoint map.

3.1.2 Generalized forces and velocities

Given any curve X(t) in SE(3), it turns out that we can interpret the tangent vector $\dot{X}(t)$ in two different ways. One as an Inertial representation and another as Body-Fixed Representation.

$$\dot{X}X^{-1} = (\dot{\Theta}\Theta^{-1}, b - \dot{\Theta}\Theta^{-1}b) \tag{3}$$

$$X^{-1}\dot{X} = (\Theta^{-1}\dot{\Theta}, \Theta^{-1}b) \tag{4}$$

The first equation gives the inertial velocity representation and second one gives body-frame representation.

After we considered generalized velocity, we consider generalized force which contained both moments and forces. The question comes is which one to denote as skew-symmetric and which one as a vector. A slight motivation could be the work done, where moment is associated with angular velocity and force is associated with velocity. Hence it can be naturally considered for the force and moments to lie in the the dual space. When X is a right translation, we can show that

$$Ad^*_X(z) = \begin{bmatrix} \Theta^T & \Theta^T[b]^T \\ 0 & \Theta^T \end{bmatrix} \begin{bmatrix} m \\ f \end{bmatrix} \tag{5}$$

resulting in that moment should be considered as a skew-symmetric matrix and force as a vector.

3.1.3 Kinetic Energy as Quadratic Form

Given that X(t) is a trajectory in SE(3), the inertial representation is given by $X^{-1}\dot{X}$. For $X^{-1}\dot{X}$ = (ω_b, v_b), the Kinetic energy can be written as

$$KE = \frac{1}{2}\omega_b^T \overline{I} \omega_b + \frac{1}{2} m v_b^T v_b$$

This motivates us to represent the Kinetic energy in the form of a Quadratic Form below:

$$QF = \begin{bmatrix} \overline{I} & 0 \\ 0 & m \cdot \mathbf{1} \end{bmatrix} \tag{6}$$

3.1.4 Product of Exponents Formulae

Suppose we have an open-chain manipulator and we are intended to relate them using Lie Group properties. We can consider the Euclidean transformation that describes the position and orientation of the ith frame in terms of (i-1)th frame; which is $f_{i-1,i} = e^{\mathbf{P}_{\mathbf{i}} x_i} M_i$. Considering all the linkages, we get

$$f(x_1, x_2, \ldots, x_n) = e^{P_1 x_1} M_1 e^{P_2 x_2} M_2 \cdot \ldots \cdot e^{P_n x_n} M_n. \tag{7}$$

Using $M^{-1}e^{P}M = e^{M^{-1}PM}$, we get,

$$f = (\prod_{i=1}^{n} e^{\mathbf{A_i}x_i})M_n \tag{8}$$

where $A_1 = P_1$, $A_2 = M_1P_2M_1^{-1}$, $A_3 = (M_1M_2)P_3(M_1M_2)^{-1}$, etc. We have another alternate way of writing f, which is

$$f = M\prod_{i=1}^{n} e^{\mathbf{B_i}x_i} \tag{9}$$

where $B_i = M^{-1}A_iM$.

To compute the matrix exponential, we can use the following rule. Let $\omega \in \text{so}(3)$ such that $\omega_1^2 + \omega_2^2 + \omega_3^2 = 1$, and $v \in \mathbb{R}^3$. Then for any $\phi \in \mathbb{R}$,

$$\exp\left(\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix}\phi\right) = \begin{bmatrix} \exp([\omega]\,\phi) & b \\ 0 & 1 \end{bmatrix} \tag{10}$$

is an element of $SE(3)$, where

$$\exp([\omega]_\times \phi) = I + \sin(\phi)\,[\omega]_\times + (1-\cos(\phi))\,[\omega]_\times^2\,, \tag{11}$$

$$b = (\phi I + (1-\cos\phi)\,[\omega]_\times + (\phi - \sin\phi)\,[\omega]_\times^2)v. \tag{12}$$

Now one can directly compute $f^{-1}\dot{f}$ in the following way.

$$f^{-1}\dot{f} = B_n\dot{x_n}e^{-B_nx_n}B_{n-1}e^{-B_nx_n}\dot{x_{n-1}} + \ldots + e^{-B_nx_n}\cdot\ldots\cdot e^{-B_2x_2}B_1e^{B_2x_2}\cdot\ldots\cdot e^{B_nx_n}\dot{x_1}, \tag{13}$$

which can also be written

$$f^{-1}\dot{f} = M^{-1}(A_n\dot{x_n} + e^{-A_nx_n}A_{n-1}e^{A_nx_n}\dot{x_{n-1}} + \ldots)M \tag{14}$$

$$= \mathrm{Ad}_{M^{-1}}(A_n\dot{x_n} + e^{-A_n x_n}A_{n-1}e^{A_n x_n}\dot{x_{n-1}} + \ldots). \tag{15}$$

3.2 Recursive formulation of Robot Dynamics

In the following recursive formula, we use forward propagation from base to tip for finding generalized velocity and backward propagation from tip to base for finding generalized force.
Given $\mathbf{J_i} \in R^{6x6}$ be the following matrix:

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

where I_i is the inertia matrix of link I with respect to its centre of mass. Now the recursive formula starts in the following way.

1) Initialization:

$$V_0 = \dot{V_0} = F_{n+1} = 0 \tag{16}$$

2) Forward recursion for i = 1 to n:

$$f_{i-1,i} = M_i e^{S_i x_i} \tag{17}$$

$$V_i = Ad_{f^{-1}_{i-1,i}}(V_{i-1}) + S_i\dot{x}_i \tag{18}$$

$$\dot{V_i} = S_i\ddot{x}_i + Ad_{f^{-1}_{i-1,i}}(\dot{V_{i-1}}) + ad_{Ad_{f^{-1}_{i-1,i}}(V_{i-1})}(S_i\dot{x}_i) \tag{19}$$

3) Backward recursion for i = n to 1:

$$F_i = Ad_{f^{-1}_{i,i+1}}(F_{i+1}) + J_i\dot{V_i} - ad^*_{V_i}(J_iV_i) \tag{20}$$

$$\tau_i = S_i^T F_i \tag{21}$$

In general, this has almost 130n - 68 scalar multiplications and 101n - 56 scalar additions leading it to an O(n) algorithm.

3.3 Lagrangian formulation of Robot Dynamics

In the Lagrangian formulation, the dynamic equations can be written as

$$\tau_k = \sum_{j=1}^{n} m_{kj}(x)\ddot{x}_j + \sum_{i=1}^{n}\sum_{j=1}^{n} \Gamma_{ijk}\dot{x}_i\dot{x}_j) + \phi_k(x), \tag{22}$$

where τ_k is the applied force/torque at joint k, m_{kj} are elements of the inertia matrix, Γ_{ijk} are the Christoffel symbols of the first kind relative to the inertia matrix representing the centrifugal and Coriolis terms, and ϕ_k are the forces due to gravity.

3.3.1 The Inertia matrix

The total kinetic energy of the system is given by the following expression:

$$T = \sum_{ij} m_{ij}(x)\dot{x}_i\dot{x}_j \tag{23}$$

After a couple of manipulations using $f_i^{-1}\dot{f}_i$ and relating it to $Ad_{M^{-1}}$ we get the following expression for Kinetic energy.

$$T_i = \frac{1}{2}\begin{bmatrix}\omega_i^T & v_i^T\end{bmatrix}\begin{bmatrix} I_i & m_i[b_i] \\ m_i[b_i]^T & m_i.\mathbf{1}\end{bmatrix}\begin{bmatrix}\omega_i \\ v_i\end{bmatrix} \tag{24}$$

This can be used to define an inner product using the quadratic form as follows.

Let $\langle \cdot, \cdot \rangle$ be an inner product on $\mathfrak{se}(3)$ defined by the quadratic form

$$T_i = \begin{bmatrix} I_i & m_i[b_i] \\ m_i[b_i]^T & m_i \end{bmatrix} \tag{25}$$

T_i can therefore be written as $\langle (\omega_i, v_i), (\omega_i, v_i) \rangle$. We also adopt the following notation:

Given $A_1, \ldots, A_n \in \mathfrak{se}(3)$ and $x_1, \ldots, x_n \in \mathbb{R}$, define the map $\mathrm{Ad}_j^i : \mathfrak{se}(3) \to \mathfrak{se}(3)$ by

$$\mathrm{Ad}_j^i(H) = \begin{cases} e^{-A_j x_j} \cdots e^{-A_i x_i} H e^{A_i x_i} \cdots e^{A_j x_j}, & i \le j \\ H, & i > j \end{cases} \tag{26}$$

Note that by this definition $\mathrm{Ad}_j^i(A_j) = A_j$. Using this notation (ω_i, v_i) from above can be written in terms of the kinematic parameters as

$$(\omega_i, v_i) = \sum_{j=1}^{i} \mathrm{Ad}_j^{i+1}(A_j). \tag{27}$$

The kinetic energy of link i is then

$$T_i = \langle \sum_{j=1}^{i} Ad_i^{j+1}(A_j, \dot{x}_j)^T Ad_i^{j+1}(A_j, \dot{x}_j) \rangle \tag{28}$$

and $T = \sum T_i$. Equating T with $\sum\sum m_{ij}(x_i, \dot{x}_i)^T (x_j, \dot{x}_j)$ and matching terms, the components of the inertia matrix are given by

$$m_{ij} = \langle \sum_{k=j}^{n} (Ad_k^{i+1}(A_i), Ad_k^{j+1}(A_j))_k \rangle \tag{29}$$

for $j \ge i$, and $m_{ij} = m_{ji}$.

We now express the inertia matrix in block-matrix form. Let $A \in \mathbb{R}^{6n \times 6n}$ be a block-diagonal matrix of the form

$$A = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n \end{bmatrix} \tag{30}$$

Then define $L \in \mathbb{R}^{6n \times 6n}$ to be the lower block-triangular matrix

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ (\mathrm{Ad}_2^1) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\mathrm{Ad}_n^1) & (\mathrm{Ad}_n^2) & \dots & 1 \end{bmatrix} \tag{31}$$

and $D_i \in \mathbb{R}^{6 \times 6}$ to be the kinetic energy quadratic form associated with link i:

$$D_i = \begin{bmatrix} I_i & m_i[b_i] \\ m_i[b_i]^T & m_i \end{bmatrix} \tag{32}$$

Note that D_i is a constant matrix with exactly 10 independent parameters. Now let $D \in \mathbb{R}^{6n \times 6n}$ be the block-diagonal matrix

$$D = \begin{bmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_n \end{bmatrix} \tag{33}$$

The inertia matrix M(x) is then given by

$$M = A^T L^T D L A \tag{34}$$

These included factoring the Inertia matrix into inertial parameters, kinematic parameters, and joint values. One can also factor them into vector and matrix quantities appearing in recursive Newton-Euler formulation using the following formulas.

$$S = \mathrm{Diag}[S_1, S_2, \ldots, S_n] \in \mathbb{R}^{6n\times 6n} \tag{35}$$

$$J = \mathrm{Diag}[J_1, J_2, \ldots, J_n] \in \mathbb{R}^{6n\times 6n} \tag{36}$$

$$G = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ \mathrm{Ad}_{f_{1,2}^{-1}} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Ad}_{f_{1,n}^{-1}} & \mathrm{Ad}_{f_{2,n}^{-1}} & \ldots & 1 \end{bmatrix} \tag{37}$$

$$Q = \mathrm{Diag}[\mathrm{Ad}_{M_1}, \mathrm{Ad}_{M_2}, \ldots, \mathrm{Ad}_{M_1,\ldots,M_n}] \tag{38}$$

Then M can be factored as

$$M = S^T G^T J G S \tag{39}$$

where $A = QS, L = QGQ^{-1}$, and $D = Q^{-1}TQ^{-1}$. This relates Newton-Euler properties with inertial, kinematic, and dynamic parameters.

3.3.2 The Coriolis Term

The Coriolis terms of the dynamic equations involve computation of Γ_{ijk}, the Christoffel symbols of the first kind relative to the metric defined by the inertia matrix:

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial m_{kj}}{\partial x_i} + \frac{\partial m_{ki}}{\partial x_j} - \frac{\partial m_{ij}}{\partial x_k}\right) \tag{40}$$

and $\Gamma_{ijk} = \Gamma_{jik}$. Lets formulate the following Propositions which will help in computing the formulae.

Proposition 1. Given the map $\mathrm{Ad}_j^i : \mathfrak{se}(3) \to \mathfrak{se}(3)$ as in Definition 2,

$$\frac{\partial}{\partial x_k}\mathrm{Ad}_j^i(B) = \begin{cases} \mathrm{Ad}_j^i([\mathrm{Ad}_{k,0}^{i-1}(A_k), B]), & i \le k \le j \\ 0, & \text{otherwise} \end{cases} \tag{41}$$

Here $[\cdot,\cdot]$ denotes the Lie bracket on $\mathfrak{se}(3)$.

Proposition 2. Given the map $\mathrm{Ad}_j^i : \mathfrak{se}(3) \to \mathfrak{se}(3)$ as in Definition 2, suppose $i \le j$. Then

$$\frac{\partial}{\partial x_l}\frac{\partial}{\partial x_k}\mathrm{Ad}_j^i(B) = \begin{cases} \mathrm{Ad}_j^l\left(\left[\mathrm{Ad}_{l-1}^k\left(\left[\mathrm{Ad}_{k-1}^i(B), A_k\right]\right), A_l\right]\right) & i \le l \le k \le j \\ \mathrm{Ad}_j^k\left(\left[\mathrm{Ad}_{k-1}^l\left(\left[\mathrm{Ad}_{l-1}^i(B), A_l\right]\right), A_k\right]\right) & i \le k \le l \le j \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

Using these formulas, the Christoffel symbols are calculated and can be segregated into three cases. Refer the paper for obtaining formula for these cases.

3.3.3 Potential Terms

Here we assume gravity is the only source of potential energy. Recall that $f_i = e^{A_i x_i} \ldots e^{A_i x_i} M_i$ describes the center of mass frame of link I relative to the inertial frame. Now consider the constant vector $\mathbf{r} = [0,0,0,1]^T$. Then the potential energy linked is given by $(m_i \mathbf{g}, f_i \mathbf{r})$. Then the total energy of the manipulator is,

$$U = \sum_{i=1}^{n} (m_i \mathbf{g}, \mathbf{f}_i \mathbf{r}). \tag{43}$$

The potential term in the equations of motion (29) is therefore given by

$$\phi_k = \frac{\partial}{\partial x_k} U. \tag{44}$$

After some manipulation, the potential term can be expressed as

$$\phi_k = \sum_{i=k}^{n} (m_i \mathbf{g} \cdot \mathbf{f}_i \cdot \mathrm{Ad}_i^{k+1}(\mathbf{A}_k)\mathbf{M}_i \mathbf{r}) \tag{45}$$

for $1 \leq k \leq i$.

3.3.4 Example

The dynamic equations for a general open chain manipulator, ignoring gravity, can be written as $\tau = M(\mathbf{x})\ddot{\mathbf{x}} + C(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}}$. For a general 2R manipulator,

$$M(\mathbf{x}) = M(x) = \begin{bmatrix} \langle A_1, A_1 \rangle_1 + \langle \mathrm{Ad}_2^2(A_1), \mathrm{Ad}_2^2(A_1) \rangle_2 & \langle \mathrm{Ad}_2^2(A_1), A_2 \rangle_2 \\ \langle \mathrm{Ad}_2^2(A_1), A_2 \rangle_2 & \langle A_2, A_2 \rangle_2 \end{bmatrix} \tag{46}$$

the $C(\mathbf{x}, \dot{\mathbf{x}})$ matrix is as follows:

$$c_{11} = \mathrm{Ad}^2(\mathbf{A}_1, \mathbf{A}_1)\mathrm{Ad}^2(\mathbf{A}_1)\dot{x}_1^2 \tag{47}$$

$$c_{12} = \mathrm{Ad}^2(\mathbf{A}_1, \mathbf{A}_1)\mathrm{Ad}^2(\mathbf{A}_1)\dot{x}_1\dot{x}_2 \tag{48}$$

$$+ \mathrm{Ad}^2(\mathbf{A}_1, \mathbf{A}_2)\dot{x}_2^2 \tag{49}$$

$$c_{21} = \mathrm{Ad}^2(\mathbf{A}_1, \mathbf{A}_1)\mathrm{Ad}^2(\mathbf{A}_1)\dot{x}_2\dot{x}_1 \tag{50}$$

$$c_{22} = 0 \tag{51}$$

3.3.5 A coordinate-free Interpretation of the Inertia matrix

The inertia matrix can be viewed as the pullback of a Riemannian metric. For an n-link revolute joint manipulator, kinematic maps from the torus to the Special Euclidean group are defined for each link. A map from the product of tori to the product of SE(3) groups is constructed. A Riemannian metric is defined as the product of individual metrics, each associated with the inertia tensor of a link. The robot's inertia matrix is the pullback of this Riemannian metric, represented in coordinates by J^TGJ, where J is the Jacobian of the map. G is block-diagonal, can be expressed using a product-of-exponentials formula, and the Jacobian is factorable.

$$f : T^m \to SE(3) \times \cdots \times SE(3)$$

$$f(x) = (f_1(x), f_2(x), \ldots, f_n(x)).$$

where we defined a Riemannian metric on the range by $g = g_1 \otimes g_2 \otimes \cdots \otimes g_n$, where each g_i is the metric defined by the generalized inertia tensor of link i.

4 Explaining some mathematical object

Let V be a vector space over a field $\mathbb{F}$. The **dual space** V^* of V consists of all linear functionals $f : V \to \mathbb{F}$ such that f is linear. This can be formally defined as:

$$V^* = \{f : V \to \mathbb{F} \mid f(\alpha x + \beta y) = \alpha f(x) + \beta f(y) \text{ for all } x, y \in V, \alpha, \beta \in \mathbb{F}\}$$

In the setting of **Lie algebras**, V is equipped with a Lie bracket $[., .]$. The dual space V^* becomes a useful tool in exploring properties like cohomology and representations of the Lie algebra. The action of the Lie algebra on its dual can be explored via the coadjoint representation, where the Lie bracket operation induces a dual action on V^*.

The Lie algebra $\mathfrak{so}(3)$ consists of all 3×3 skew-symmetric matrices. A general element X of $\mathfrak{so}(3)$ can be expressed as:

$$X = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

where x, y, z are real numbers. This matrix representation corresponds to the cross product operation on $\mathbb{R}^3$.

The dual space $\mathfrak{so}(3)^*$ consists of linear functionals on $\mathfrak{so}(3)$. Since $\mathfrak{so}(3)$ is three-dimensional, so is $\mathfrak{so}(3)^*$. A basis for $\mathfrak{so}(3)^*$ can be thought of as functionals that evaluate the coefficients x, y, and z of the matrix X.

An element f in $\mathfrak{so}(3)^*$ can be uniquely determined by its values on a basis of $\mathfrak{so}(3)$, say E_1, E_2, E_3, where:

$$E_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The functionals corresponding to these basis elements could be defined as:

$$f_1(E_1) = 1, \quad f_1(E_2) = 0, \quad f_1(E_3) = 0,$$

$$f_2(E_1) = 0, \quad f_2(E_2) = 1, \quad f_2(E_3) = 0,$$

$$f_3(E_1) = 0, \quad f_3(E_2) = 0, \quad f_3(E_3) = 1.$$

These functionals f_1, f_2, f_3 form a basis for $\mathfrak{so}(3)^*$ and can be used to analyze properties like the coadjoint orbits of the Lie algebra.

5 Working out with an example

We consider a robotic arm with n joints, where each joint angle θ_i can be associated with an element of a Lie group, and the transformations between links correspond to operations in this group. The recursive dynamics formulation applies as follows:

The configuration space of the arm is modeled by the Lie group G, and the corresponding Lie algebra $\mathfrak{g}$ with the tangent space at the identity, representing the velocities of the joints. We denote the exponential map from $\mathfrak{g}$ to G by $\exp : \mathfrak{g} \to G$, which relates the angular velocities to the rotations in G.

5.1 Recursive Dynamics Formulation on G

Given the inertia matrix I_i for each link in the Lie algebra representation, and using the adjoint map Ad, which relates the velocities and forces in the Lie algebra to those in different configurations in G, we can compute the dynamics of the robotic arm:

1. **Initialization:**

$$V_0 = \dot{\theta}_0 = 0, \qquad \text{(Zero velocity at base)}$$
$$F_{n+1} = 0 \qquad \text{(No external force at tip)}$$

2. **Forward Recursion:** For $i = 1$ to n:

$$\omega_{i-1} = \mathrm{Ad}_{\exp(-\theta_{i-1})}(\omega_{i-2}) + \dot{\theta}_{i-1}\hat{e}_{i-1}$$
$$V_i = \mathrm{Ad}_{\exp(-\theta_i)}(V_{i-1}) + S_i\dot{\theta}_i$$
$$\dot{V}_i = S_i\ddot{\theta}_i + \mathrm{Ad}_{\exp(-\theta_i)}(\dot{V}_{i-1}) + \mathrm{ad}_{\dot{\theta}_{i-1}\hat{e}_{i-1}}(V_{i-1})$$

where $\hat{e}_i$ is the unit twist for joint i, and S_i is the motion subspace matrix for joint i.

3. **Backward Recursion:** For $i = n$ to 1:

$$F_i = \mathrm{Ad}^*_{\exp(\theta_{i+1})}(F_{i+1}) + J_i \dot{V}_i$$
$$\tau_i = S_i^T F_i$$

   where Ad^* is the coadjoint action, representing the dual adjoint action of G on the dual space of $\mathfrak{g}$.

Here, J_i is the inertia matrix of link i, τ_i is the torque at joint i, and F_i is the force on link i. This recursive algorithm efficiently computes the torques required at each joint to achieve a desired motion, characterizing the dynamic behavior of the robotic arm within the framework of Lie groups.

6 Drawbacks and Positives of the Paper

6.1 Positives

- Lesser need of complex definitions and notational conventions
- Clear connection between geometric ideas and aspects of robotics
- Geometric understanding of Newton-Euler formulation and Lagrangian formulation.
- A Recursive O(n) algorithm is developed to link velocities, acceleration, and generalized forces with Lie algebra operations in SE(3)
- Explicit formula representation using Lagrangian formulation leading to gradients that can be obtained easily and expressions that can be factored efficiently.

6.2 Drawbacks

- We could not get a better iterative algorithm to find the generalized velocity and generalized force.
- The explicit solution obtained in Lagrangian formulation is still very complicated to implement and factor out.