Recursive Algorithm for Second-Order Time Derivatives

Mihir Paygude[1]
[1] RIC, DFKI, Bremen, [2] University of Twente, Netherlands

Introduction

Serial Elastic Actuators (SEAs) have significant advantages over traditional rigid actuators such as;[1]

- Force control and safety : Compliance due springs allows for accurate force sensing and control
- Protection against Shock Loads : The spring can absorb and dissipate excessive forces.
- Low Noise : Compliance in springs also reduces noise and vibrations.
- SEAs provide better accuracy in force and position control than rigid actuators
- Smooth movements : SEAs provide smooth and natural movements which are essential for Humanoid robots

SEAs are predominantly utilised in Serial Manipulators. Implementation of SEAs for Parallel Kinematic Manipulators (PKMs) is yet to be pursued.

Challenges in implementing SEAs in PKM robots:

- Tuning and Control : Achieving the right balance between compliance and control stiffness to ensure stability and performance.
- Complex Control Algorithm : Control and coordination in humanoid robots with SEAs requires sophisticated control algorithms, including impedance control and trajectory optimization.

Contribution : A computationally efficient recursive algorithm using Lie-Group formulation and exploiting the special topology of PKM is presented. This enables better control of the PKM which ensures stability and better performance.

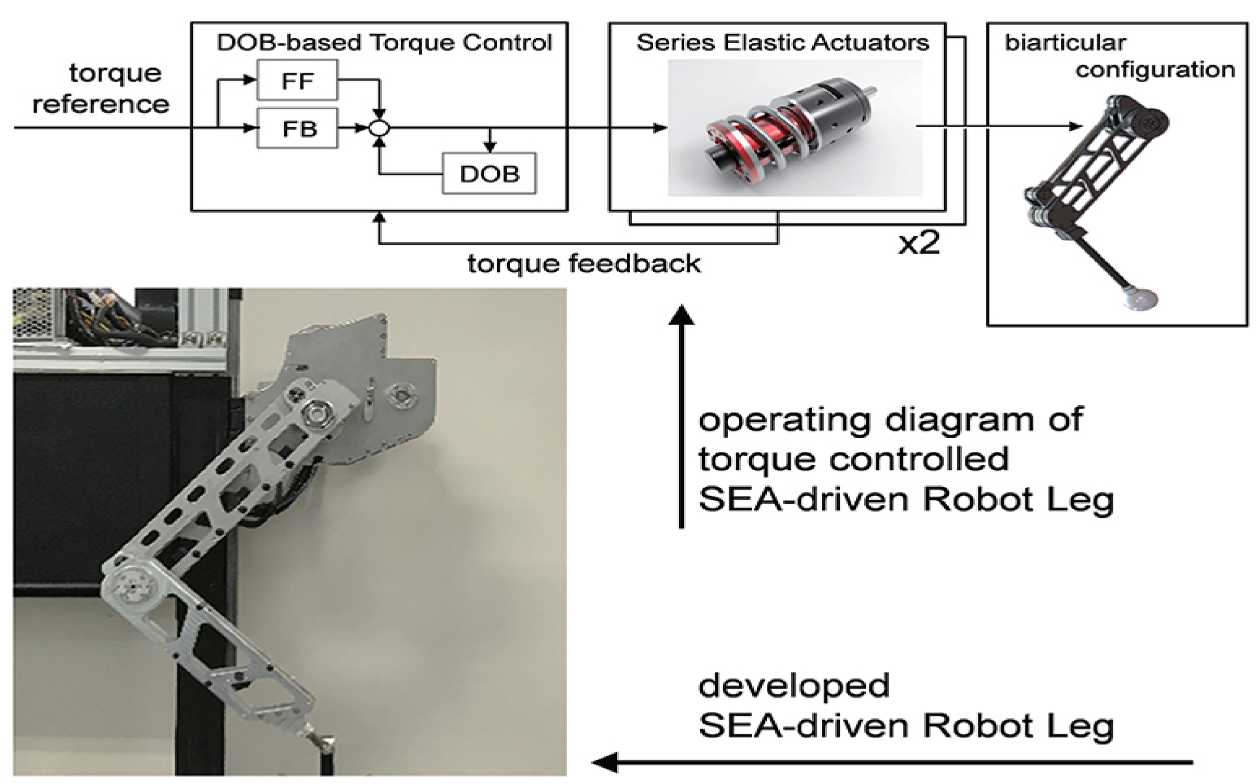

Figure: a)Implementation of a Series Elastic Actuator for a Serial Chain Mechanism i.e. Robot Leg using a force controller

Tree-topology system of PKM

Tree-topology system is implemented in which the simplified model of the platform is shown such that cut-joints are eliminated for analysing multi-body dynamics.

- Platform is attached to one leg L (treated as a leaf), while $L-1$ limbs are cut-off from the platform.
- Each limb is treated as one serial kinematic chain
- The tree consists of L congruent sub-graphs, this enables tailored kinematics modelling for each sub-graph in terms of platform motion.
- All technical joints are modelled as combination of 1-DOF joints.[2]

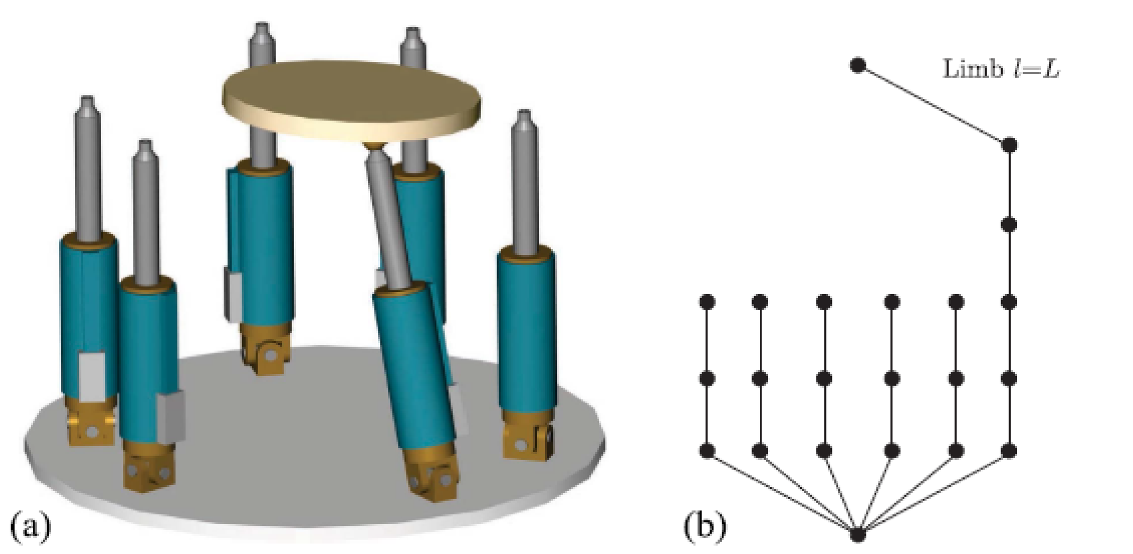

Figure: a)Tree-Topology system of the Stewart-Gough Platform such that one limb is connected to platform. b) Spanning Tree

Dynamics of SEA-PKM

The equations of motion (EOM) that govern the PKM dynamics in task space and actuator dynamics are given as;

$$M_t\dot{V}_t + C_tV_t + W_t^{grav} = J_{IK}^T u \quad (1)$$

$$M_m\ddot{q}_m = \tau - u \quad (2)$$

here M_t is the Generalized Mass Matrix, C_t is Coriolis Matrix, W_t^{grav} are Generalized Gravity Forces, M_m is Diagonal Matrix of Reduced inertia moment, q_m is vector of n motor co-ordinates, τ is vector of actuator forces and J_{IK} is inverse kinematics Jacobian

The elastic force is given as $u = K(q_m - \vartheta_a)$. K is the SEA compliance and ϑ_a denotes co-ordinate vector. To devise a control scheme for these EOM we need;

1. Evaluation of τ for feed-forward control.
2. To solve (2) for τ, we need to compute second-order time derivative of q_m.
3. We obtain q_m, by computing u from (1).
4. To this end, we need second-order time derivative of u

Joint space formulation of EOM of each limb

The EOM of limb l for any multi-body system in standard form is given as:

$$\bar{M}_{(l)}\ddot{\vartheta}_{(l)} + \bar{C}_{(l)}\dot{\vartheta}_{(l)} + \bar{Q}^{grav}_{(l)} = \bar{Q}_{(l)} \quad (3)$$

where, $\bar{M}_{(l)}$ is the generalized Mass matrix, $\bar{C}_{(l)}$ the generalized Coriolis matrix. $\bar{Q}^{grav}_{(l)}$ is generalized forces due to gravity and $\bar{Q}_{(l)}$ are all the applied forces due to actuator forces. We use this formulation to evaluate u and it's time derivatives given by ordinary inverse dynamics formulation;

$$u = J_{IK}^{-T}\sum_{l=1}^{L}\bar{Q}_{(l)}(\vartheta,\dot{\vartheta},\ddot{\vartheta}) \quad (4)$$

$$\dot{u} = J_{IK}^{-T}\left(\sum_{l=1}^{L}(\dot{\bar{Q}}_{(l)}) - \dot{J}_{IK}^{T}u\right) \quad (5)$$

$$\ddot{u} = J_{IK}^{-T}\left(\sum_{l=1}^{L}(\ddot{\bar{Q}}_{(l)}) - \ddot{J}_{IK}^{T}u - 2\dot{J}_{IK}^{T}\dot{u}\right) \quad (6)$$

Forward and Inverse Kinematics of Limbs

The forward kinematics gives the platform twist V_p in terms of the joint co-ordinates of the limb l as;

$$V_p = J_{p(l)}\dot{\vartheta}_{(l)} \quad (7)$$

here $J_{p(l)}$ is the forward kinematics Jacobian of limb l.

The inverse kinematics on velocity and acceleration levels, which also imposes constraints on the joint-coordinates is given as;

$$\dot{\vartheta}_{(l)} = J_{p(l)}^{-1}V_p \quad \text{and} \quad \ddot{\vartheta}_{(l)} = J_{p(l)}^{-1}\dot{V}_p + \dot{J}_{p(l)}^{-1}V_p \quad (8)$$

Fourth-Order Forward/Inverse Kinematics

The second-order inverse kinematics requires the evaluation of 4-th order inverse kinematics of each limb of the mechanism (i.e. computing $\vartheta, \ldots, \ddddot{\vartheta}$ from the platform twist derivatives $V_p, \ldots, \dddot{V}_p$) and the forward kinematics (computing body twists $V_{i(l)}, \ldots, \dddot{V}_{i(l)}$ from $\vartheta, \ldots, \ddddot{\vartheta}$). This computation is done in a single run;

- INPUT : ϑ, V_p
- OUTPUT : $\dot{\vartheta}, \ddot{\vartheta}, \dddot{\vartheta}, \ddddot{\vartheta}, V_i, \dot{V}_i, \ddot{V}_i, \dddot{V}_i$.

Inverse Dynamics Algorithm for $\bar{Q}_{(l)}$

To evaluate second-order time derivative of u given by (4) we need to evaluate time derivatives of $\bar{Q}_{(l)}$ for which we require the 3rd-order time derivatives of body twists $V_{i(l)}$ (these are obtained from forward kinematics as shown above). Hence in this algorithm we get;

- INPUT : $V_i, \dot{V}_i, \ddot{V}_i, \dddot{V}_i$
- OUTPUT : $\bar{Q}_{(l)}, \dot{\bar{Q}}_{(l)}, \ddot{\bar{Q}}_{(l)}$.

Example and Simulation Results

The proposed algorithm was applied to 6-DOF U$\bar{P}$S Stewart-Gough Platform, which is used in torso of a Recupera-Reha exoskeleton. [3]

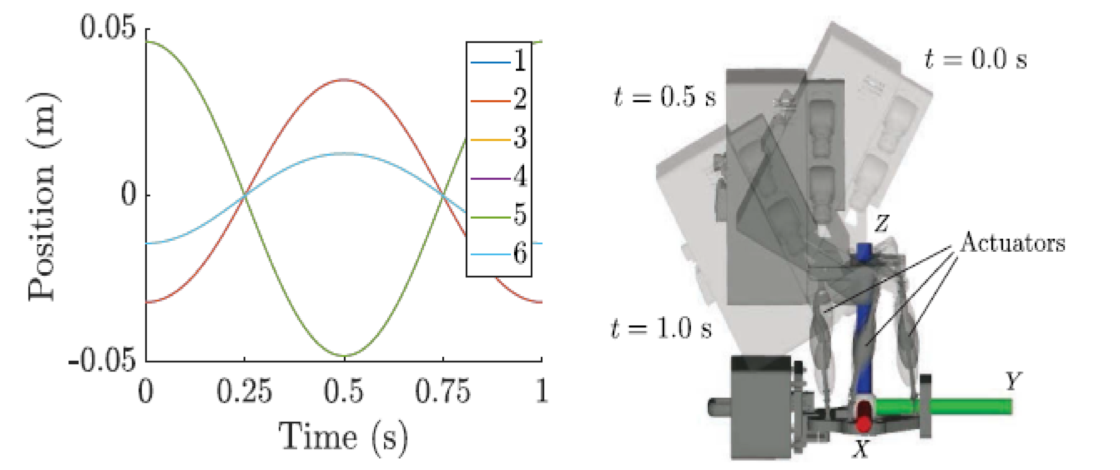

Figure: a) Inverse Kinematics Solution. b) Animation of Stewart-Gough Platform of Recupera-Reha Exoskeleton

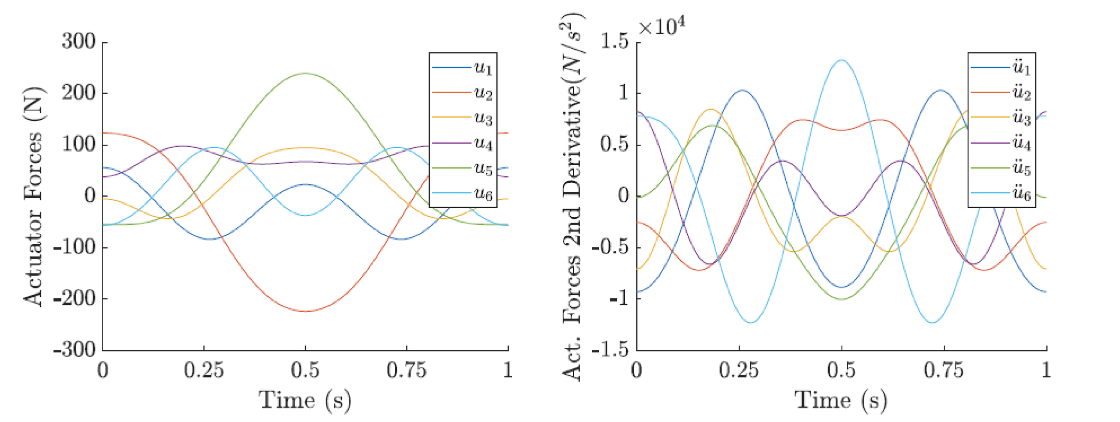

Figure: a) 1st-Order Time derivative of actuator forces $u(t)$.
b) 2nd-Order Time derivative of actuator forces $u(t)$

Conclusion

- Computationally efficient recursive algorithm is presented whose performance is sufficient for real-time control application.
- A Lie-group formulation allows us to have compact invariant expressions.
- We are able to separate the overall kinematics of the PKM and the individual limbs. This leads to model simplification in case of series-parallel mechanism.

Acknowledgment

This work was supported by the federal state of Bremen for setting up the Underactuated Robotics Lab under Grant 201-342-04-2/2021-4-1. This work was also supported by Austria COMET-K2 program and VeryHuman (FKZ 01/W20004) funded by the Federal Ministry of Austria and Federal Ministry of Education and Research (BMBF) of Germany respectively.

References

[1] Jerry Pratt, Ben Krupp, and Chris Morse. "Series elastic actuators for high fidelity force control". In: *Industrial Robot: An International Journal* 29 (June 2002), pp. 234–241. DOI: 10.1108/01439910210425522.

[2] Andreas Müller, Shivesh Kumar, and Thomas Kordik. "A Recursive Lie-Group Formulation for the Second-Order Time Derivatives of the Inverse Dynamics of Parallel Kinematic Manipulators". In: *IEEE Robotics and Automation Letters* 8.6 (2023), pp. 3804–3811. DOI: 10.1109/LRA.2023.3267005.

[3] Shivesh Kumar et al. "Modular Design and Decentralized Control of the Recupera Exoskeleton for Stroke Rehabilitation". In: *Applied Sciences* 9.4 (2019). ISSN: 2076-3417. DOI: 10.3390/app9040626. URL: https://www.mdpi.com/2076-3417/9/4/626.