A dual degree project presentation on

A Study on the Lagrangian Formulation of Dynamics with Applications to Control of Parallel Manipulators

by

Mr Akhil Sathuluri

Adviser: Dr Sandipan Bandyopadhyay

Department of Engineering Design,
Indian Institute of Technology Madras, Chennai - 600 036.

Introduction

Formulations

Comparison

Control of SRSPM

Conclusion

Problem Statement

- Modelling: Obtain a Lagrangian dynamics model which enables fast computer simulation without any approximation of the system

Problem Statement

- Modelling: Obtain a Lagrangian dynamics model which enables fast computer simulation without any approximation of the system
- Implementation: Simulate and compare various Lagrangian dynamic formulations

Problem Statement

- Modelling: Obtain a Lagrangian dynamics model which enables fast computer simulation without any approximation of the system
- Implementation: Simulate and compare various Lagrangian dynamic formulations
- Utility: Solve a trajectory tracking problem to demonstrate the improvement in speed of computation

Motivation

- The closed-loop architecture of a parallel manipulator gives rise to sizeable symbolic expressions relating the passive and the actuated variables [Lebret et al., 1993].

Motivation

- The closed-loop architecture of a parallel manipulator gives rise to sizeable symbolic expressions relating the passive and the actuated variables [Lebret et al., 1993].
- These have a significant effect on the simulation time of the system [Nguyen et al., 1993].

Motivation

- The closed-loop architecture of a parallel manipulator gives rise to sizeable symbolic expressions relating the passive and the actuated variables [Lebret et al., 1993].
- These have a significant effect on the simulation time of the system [Nguyen et al., 1993].
- The model is often idealised to meet real-time execution constraints [Lee et al., 2003, Davliakos and Papadopoulos, 2008].

Motivation

- Non-obvious advantage of a Newton-Euler model for a lower order system [Silver, 1982].

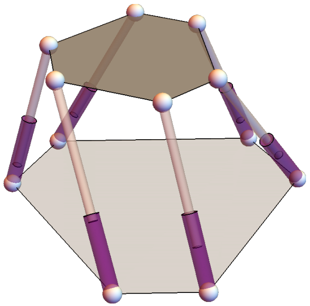
Motivation

- Non-obvious advantage of a Newton-Euler model for a lower order system [Silver, 1982].
- Differential geometric perspective of the problem is brought in Lagrangian formulation [Lewis, 2007].

Motivation

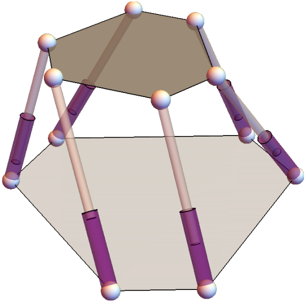
- Non-obvious advantage of a Newton-Euler model for a lower order system [Silver, 1982].
- Differential geometric perspective of the problem is brought in Lagrangian formulation [Lewis, 2007].
- Enable controllability study and non-linear controller design [Choudhury and Ghosal, 2000].

Semi-Regular Stewart Platform Manipulator (SRSPM)



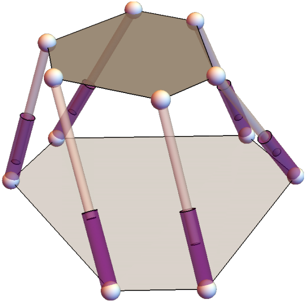
- The reference system for all the dynamic simulations is taken to be SRSPM [Stewart, 1965].

Semi-Regular Stewart Platform Manipulator (SRSPM)



- The reference system for all the dynamic simulations is taken to be SRSPM [Stewart, 1965].
- It is a 6-degree-of-freedom platform type manipulator.

Semi-Regular Stewart Platform Manipulator (SRSPM)

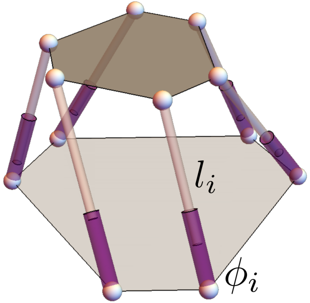


- The reference system for all the dynamic simulations is taken to be SRSPM [Stewart, 1965].
- It is a 6-degree-of-freedom platform type manipulator.
- SRSPM has applications in simulators and machine tools [Freeman et al., 1995, Pradipta et al., 2013, Lebret et al., 1993].

Simulation set-up

Consider the SRSPM manipulator falling freely from a symmetric (no relative rotation between the moving and the base platforms) initial condition only under the influence of gravity. The centre of mass of the top platform is at the height of $z = 1.1$ m at time $t = 0$ s and the simulation is performed for 1 s.

Configuration-space dynamic model



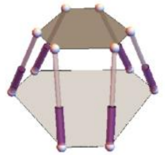
$$\boldsymbol{q} = [\boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top]^\top$$

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \boldsymbol{0}$$
$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{Q}^{\mathrm{nc}} + \boldsymbol{J}_{\boldsymbol{\eta q}}^\top \boldsymbol{\lambda}$$

Table: Sizes of the coefficient matrices of configuration-space dynamic model of SRSPM

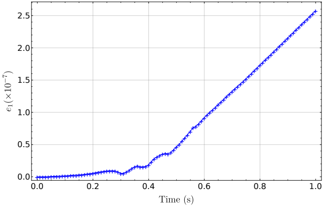
	Size (MB)	
Expression	`C++`	`Mathematica`
$\boldsymbol{M}$	0.900	13.369
$\boldsymbol{C}$	103.4	748.207

Configuration-space dynamic model (contd. . .)

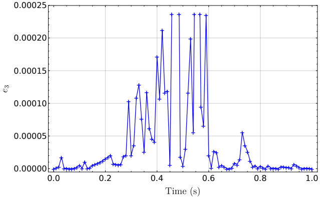


- Compilation and execution of $\boldsymbol{C}$ matrix in `C++` is expensive
- The simulation is performed using the numerical Runge-Kutta-Fehlberg solver

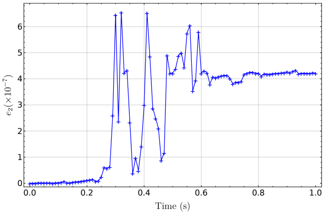
Validation of the model [Muralidharan et al., 2018]



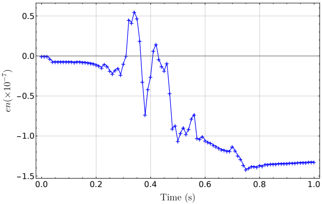
$$e_1 = \max(|\boldsymbol{\eta}(\boldsymbol{q})|)$$



$$e_3 = \max(|\boldsymbol{J}_{\boldsymbol{\eta q}}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}|)$$



$$e_2 = \max(|\boldsymbol{J}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}|)$$



$$\delta E = \frac{E(t) - E(0)}{E(0)} \times 100$$

Validation of the model (cont. . .)

Consistency checks might not identify:

- Representation errors: Constant offsets in orientation, location of CoM etc.

Validation of the model (cont. . .)

Consistency checks might not identify:

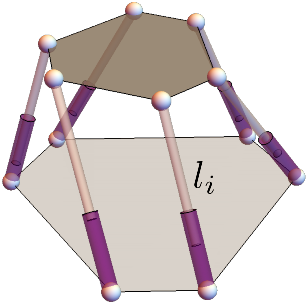
- Representation errors: Constant offsets in orientation, location of CoM etc.
- Inertia mismatch: Different physical parameters of the manipulator like mass, inertia, link lengths etc.

Validation of the model (cont...)

Consistency checks might not identify:

- Representation errors: Constant offsets in orientation, location of CoM etc.
- Inertia mismatch: Different physical parameters of the manipulator like mass, inertia, link lengths etc.
- Numerical inaccuracies: Using a mathematically consistent pseudoinverse in computations.

Actuator-space dynamic model



$\boldsymbol{\theta}$

- $\boldsymbol{M_\theta}\ddot{\boldsymbol{\theta}} + \boldsymbol{C_\theta}\dot{\boldsymbol{\theta}} + \boldsymbol{G_\theta} = \boldsymbol{Q}_a^{\mathrm{nc}}$
- Actuator-space dynamic model is derived by a transformation of coordinates from the configuration-space model
- Hence the coefficient matrices are functions of $\boldsymbol{q}$
- Configuration-space variables are required for simulation

Root-tracking of forward kinematic solutions

Nearest-neighbour method

- Computationally intensive

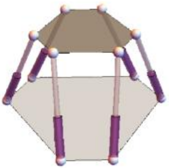
Initial value problem method

- Tracks only one branch of solutions
- Satisfaction of loop-closure constraint is not ensured

Newton-Raphson method

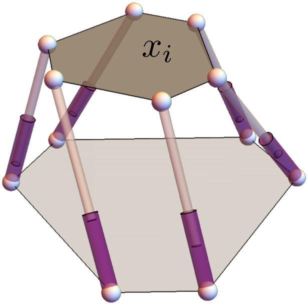
- Imposing the constraint explicitly
- Control the error in the loop-closure equations

Actuator-space dynamic model (contd. . .)



$\boldsymbol{J}_{\eta\phi}$ Jacobian matrix is used for mapping the configuration-space to the actuator-space dynamics, which becomes degenerate and hence the simulation cannot proceed after $t = 0.48$ s

Task-space dynamic model



$\boldsymbol{x}$

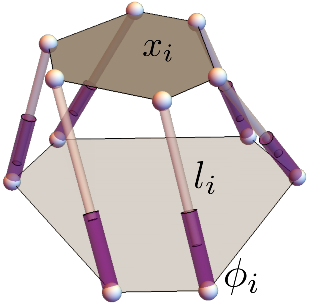
Table: Sizes of the coefficient matrices used in dynamics modelled in the task-space of the manipulator

Expression	Size (MB)
$\boldsymbol{M}$	369.044
$\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}_i}, i = 1, 2, \ldots 6$	18531.400

Formulation of task-space dynamics is computationally expensive [Lebret et al., 1993]

$$C_{ij} = \frac{1}{2} \sum_{k=1}^{18} \left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot{q}_k \quad (1)$$

Extended-configuration-space



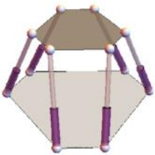
$$\boldsymbol{q}_e = [\boldsymbol{x}^\top, \boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top, \boldsymbol{\psi}^\top]^\top$$

Prevent the use of mapping between the variable sets

Table: Size of the coefficient matrices used in dynamics modelled in the extended-configuration-space

	Size (KB)	
Expression	`C++`	`Mathematica`
$\boldsymbol{M}$	4.100	35.515
$\boldsymbol{C}$	15.500	100.984

Mapping to lower dimensional spaces



Extended-configuration-space to task-space $\{\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}\} \mapsto \{\boldsymbol{x}\}$
This mapping is enabled by the $\boldsymbol{J}_{\boldsymbol{\eta q}}$ Jacobian matrix, degeneracies of which are studied in [Muralidharan et al., 2018].

Extended-configuration-space to actuator-space $\{\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}\} \mapsto \{\boldsymbol{\theta}\}$
This map is enabled by $\boldsymbol{J}_{\boldsymbol{\eta x \phi}}$ which is observed to be degenerate and needs further investigation.

Execution time

The execution time taken by each of the formulation for a free-fall simulation is used for comparison. All the models are simulated for $t = 1$ s.

Table: Computational time taken for the free-fall simulation in `C++`

Formulation	**Real time (s)**	**Simulation time (s)**	**Compilation time (min)**
Configuration space	1.000	0.223	3.259
Actuator space	0.480	0.193	3.229
Extended mapped to task-space	1.000	0.036	0.255

Numerical error

Table: Comparison of maximum error in the loop-closure equation

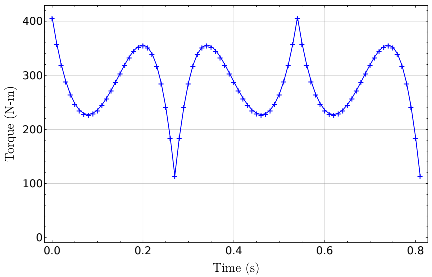
Formulation	**Order of the error** (e_1)
Configuration-space	10^{-7}
Actuator-space	10^{-13}
Task-space	10^{-15}
Extended-configuration-space mapped to task-space	10^{-16}
Extended-configuration-space mapped to actuator-space	10^{-16}

The extended-configuration-space performs better in terms of both time taken and accuracy of the simulation.

Heave motion generation for 6-RSS manipulator

Manipulator parameters given at the end

Torque requirement for moving from $z = 0.3$ m to $z = 0.4$ m with a constraint on the initial and final velocities to be 0 and the bounds on acceleration is $\pm 0.8g$.



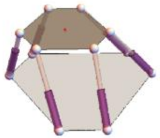
Trajectory tracking control

The extended-configuration-space mapped to actuator-space dynamic model is used to implement feedback linearisation, a model-based controller, augmented by PD control to:

1. Investigate the order of error in following a given path
2. Investigate the error response for high frequency motion requirement

The controller gains are tuned to be $K_p = 2000$ and $K_d = 2\sqrt{2000}$ to meet a settling time constraint of 0.2 s and limit the control inputs to be ≤ 10000 N.

Following Lissajous curve

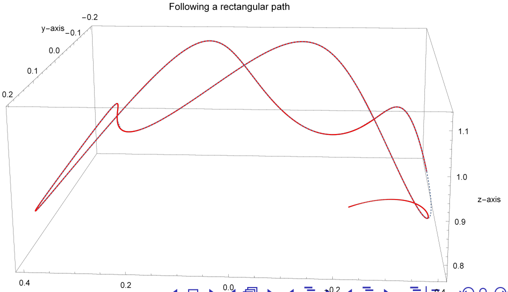


Simulation time: 3.14 s
Run-time: 1.7 s
Order of error after settling time: 10^{-8}

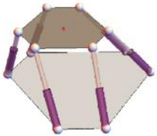
Parametrisation of the curve,

$$x(t) = x_p \ \sin(\omega_x t - \delta_x)$$
$$y(t) = y_p \ \sin(\omega_y t - \delta_y)$$
$$z(t) = 1 + z_p \ \sin(\omega_z t - \delta_z)$$

Followed (red, solid line) vs actual path (blue, dotted lines)

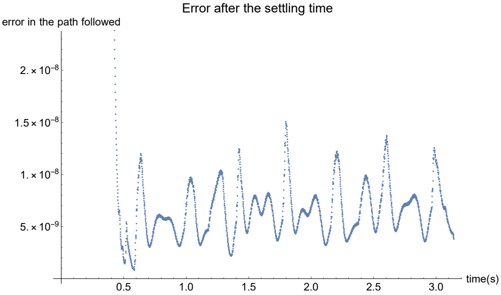


Following Lissajous curve

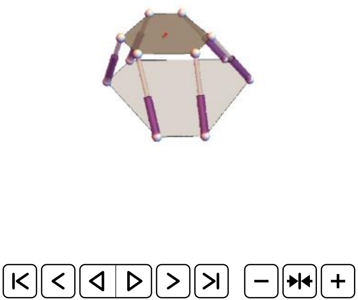


Simulation time: 3.14 s
Run-time: 1.7 s
Order of error after settling time: 10^{-8}

Plot showing the path error, defined as the absolute difference between the current and the desired positions vs. time. Only the part after the settling time is highlighted.

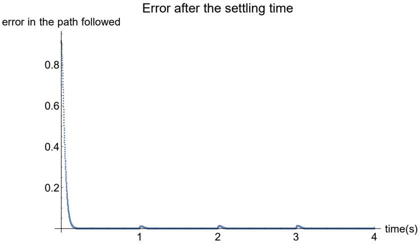


Following the rectangular path



Rectangle of dimensions 1 m×0.5 m

Plot showing the path error, defined as the absolute difference between the current and the desired positions vs. time, which converges asymptotically to 0.

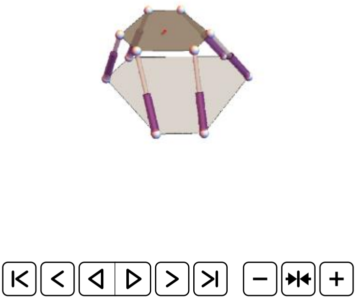


Simulation time: 4 s

Run-time: 3.3 s

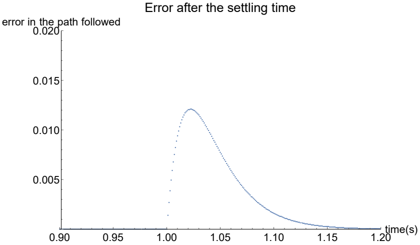
Order of error: 10^{-4}

Following the rectangular path

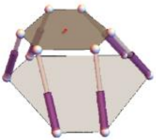


Simulation time: 4 s
Run-time: 3.3 s
Order of error: 10^{-4}

A closer look at the errors at the vertex

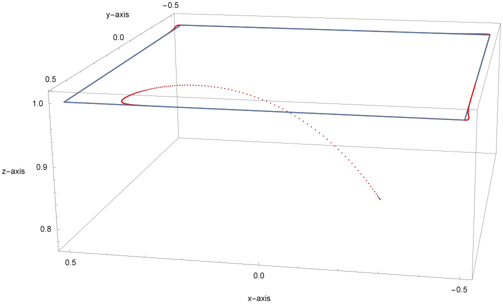


Following the rectangular path

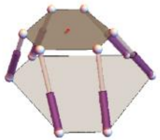


Simulation time: 4 s
Run-time: 3.3 s
Order of error: 10^{-4}

Tracked (red solid line) vs desired path (blue dashed line)

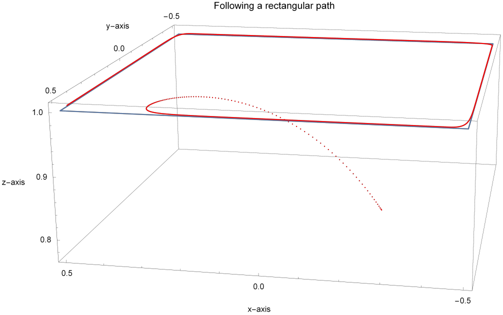


Following the rectangular path



Simulation time: 2 s
Run-time: 2.2 s
Order of error: 10^{-3}

Tracked (red solid line) vs desired path (blue dashed line)



Summary

- Various Lagrangian formulations were implemented in `C++` and are compared in terms of execution speed and relative sizes of their coefficient matrices

Summary

- Various Lagrangian formulations were implemented in `C++` and are compared in terms of execution speed and relative sizes of their coefficient matrices

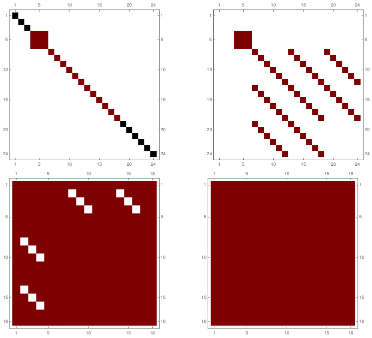
- Extended-configuration-space dynamics model is formulated collecting all the variables associated with the elements of the system to avoid mapping and hence reduce computation time

Summary

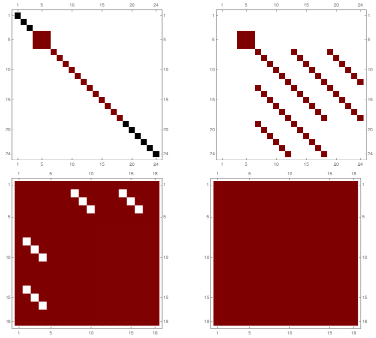
- Various Lagrangian formulations were implemented in `C++` and are compared in terms of execution speed and relative sizes of their coefficient matrices
- Extended-configuration-space dynamics model is formulated collecting all the variables associated with the elements of the system to avoid mapping and hence reduce computation time
- Trajectory-tracking controller is implemented to demonstrate the same

Possible extensions

- Sparsity in the coefficient matrices can be exploited

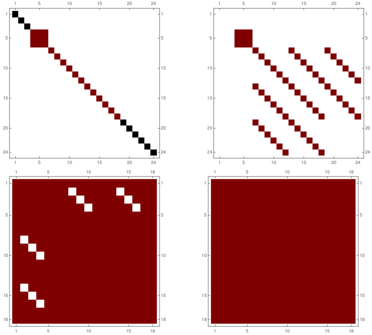


Possible extensions



- Sparsity in the coefficient matrices can be exploited
- Operational space formulation can further make the control implementation faster

Possible extensions



- Sparsity in the coefficient matrices can be exploited
- Operational space formulation can further make the control implementation faster
- Despite the map between extended-configuration-space to actuator space being degenerate, solutions can be obtained due to numerical perturbations, which needs to be further studied

Thank you all for your attention!

Any questions/comments?

Choudhury, P. and Ghosal, A. (2000).
Singularity and controllability analysis of parallel manipulators and closed-loop mechanisms.
Mechanism and Machine Theory, 35(10):1455–1479.

Davliakos, I. and Papadopoulos, E. (2008).
Model-based control of a 6-dof electrohydraulic Stewart-Gough platform.
Mechanism and machine theory, 43(11):1385–1400.

Freeman, J., Watson, G., Papelis, Y., Lin, T., Tayyab, A., Romano, R., and Kuhl, J. (1995).
The iowa driving simulator: An implementation and application overview.
Technical report, SAE Technical Paper.

Lebret, G., Liu, K., and Lewis, F. L. (1993).
Dynamic analysis and control of a Stewart platform manipulator.
Journal of Robotic Systems, 10(5):629–655.

Lee, S.-H., Song, J.-B., Choi, W.-C., and Hong, D. (2003).
Position control of a Stewart platform using inverse dynamics control with approximate dynamics.
Mechatronics, 13(6):605–619.

Lewis, A. D. (2007).
Is it worth learning differential geometric methods for modeling and control of mechanical systems?
Robotica, 25(6):765–777.

Muralidharan, V., Mamidi, T. K., Guptasarma, S., Nag, A., and Bandyopadhyay, S. (2018).
A comparative study of the configuration-space and actuator-space formulations of the lagrangian dynamics of parallel manipulators and the effects of kinematic singularities on these.
Mechanism and Machine Theory, 130:403 – 434.

Nguyen, C. C., Antrazi, S. S., Zhou, Z.-L., and Campbell Jr, C. E. (1993).

Adaptive control of a Stewart platform-based manipulator.
Journal of Robotic systems, 10(5):657–687.

Pradipta, J., Klünder, M., Weickgenannt, M., and Sawodny, O. (2013).
Development of a pneumatically driven flight simulator Stewart platform using motion and force control.
In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 158–163. IEEE.

Silver, W. M. (1982).
On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators.
The International Journal of Robotics Research, 1(2):60–70.

Stewart, D. (1965).
A platform with six degrees of freedom.
Proceedings of the Institution of Mechanical Engineers, 180(1):371–386.

SRSPM parameters

Parameter	Value
r_b	1 (m)
r_t	0.5803 (m)
γ_b	0.2985 (rad)
γ_t	0.6573 (rad)
I_{tpxx}	1.17172 (kg m^2)
I_{tpyy}	1.17172 (kg m^2)
I_{tpzz}	2.34310 (kg m^2)
I_{laxx}	2.1272 (kg m^2)
I_{layy}	2.1272 (kg m^2)
I_{lazz}	0.00173 (kg m^2)
I_{lbxx}	0.02431 (kg m^2)
I_{lbyy}	0.02431 (kg m^2)
I_{lbzz}	0.00040 (kg m^2)
m_p	0.20339 (kg)
$m_{bi}, i = (1, \ldots, 6)$	11.3404 (kg)
$m_{ai}, i = (1, \ldots, 6)$	1.15719 (kg)
$l_{bi}, i = (1, \ldots, 6)$	0.5 (m)
$l_{ai}, i = (1, \ldots, 6)$	1.5 (m)

6-RSS manipulator parameters

Parameter	Value
r_b	$1\ (m)$
r_t	$0.6\ (m)$
γ_b	$0.8080\ (rad)$
γ_t	$0.4040\ (rad)$
I_{tpxx}	$1.1717\ (kgm^2)$
I_{tpyy}	$1.1717\ (kgm^2)$
I_{tpzz}	$2.3431\ (kgm^2)$
I_{lxx}	$0.0283\ (kgm^2)$
I_{lyy}	$0.0283\ (kgm^2)$
I_{lzz}	$6.325{\times}10^{-6}\ (kgm^2)$
I_{rxx}	$0.1764\ (kgm^2)$
I_{ryy}	$0.1764\ (kgm^2)$
I_{rzz}	$5.400{\times}10^{-6}(kgm^2)$
m_p	$200.2033\ (kg)$
$m_{li}, i = (1, \ldots, 6)$	$0.5060\ (kg)$
$m_{ri}, i = (1, \ldots, 6)$	$0.4320\ (kg)$
$l_{li}, i = (1, \ldots, 6)$	$0.8200\ (m)$
$l_{ri}, i = (1, \ldots, 6)$	$0.7000\ (m)$

Details of the tracked 3D-Lissajous curve

Parameter	Value
x_p	0.400 (m)
y_p	0.200(m)
z_p	0.133 (m)
ω_x	2 (rad[1]/s)
ω_y	8 (rad/s)
ω_z	4 (rad/s)
δ_x	$\pi/2$ (rad)
δ_y	0 (rad)
δ_z	$\pi/2$ (rad)

[1] rad means radians