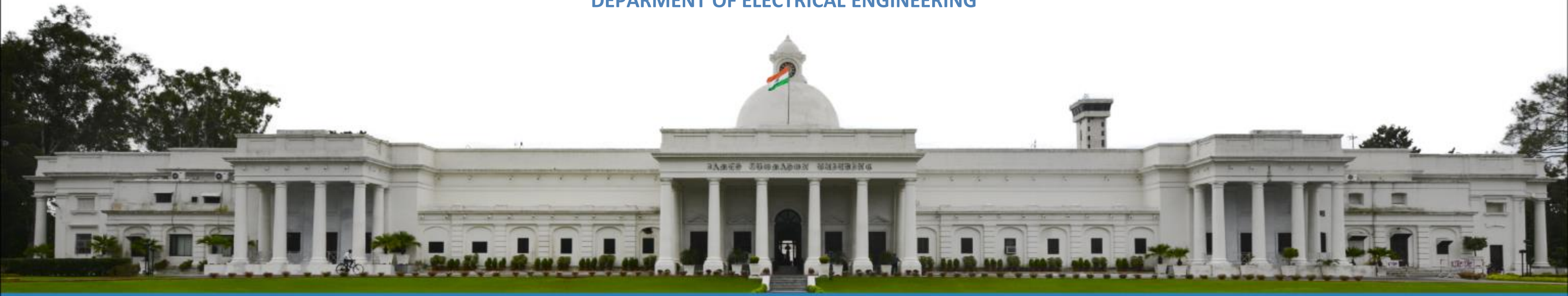
IIT ROORKEE

NPTEL ONLINE CERTIFICATION COURSE

Prof. FELIX ORLANDO MARIA JOSEPH

DEPARMENT OF ELECTRICAL ENGINEERING



Neural Control of a Hand Exoskeleton

Outline

1. Introduction

2. KSOM Architecture

3. Comparison with Inverse Kinematics

4. Conclusion

Introduction

- The Kinematic control of a robot is difficult since → we require inverse kinematic relationship to actuate the robot to the desired position.

- Model based methods require accurate knowledge of the robot and, also the computation of the inverse of the Jacobian matrix to obtain the joint angle.

- In the case of redundant manipulator, the Jacobian is a non-square matrix and, hence, the pseudo-inverse is used to compute the joint angle.

- Alternatively, neural network based approaches <u>avoid the necessity of estimating the forward kinematics</u> and also the <u>computation of the pseudo-inverse of the Jacobian.</u>

- The training of the neural network in supervisory mode to control the robot is infeasible, since, the data which represents the joint angle corresponding to the desired position is not available.

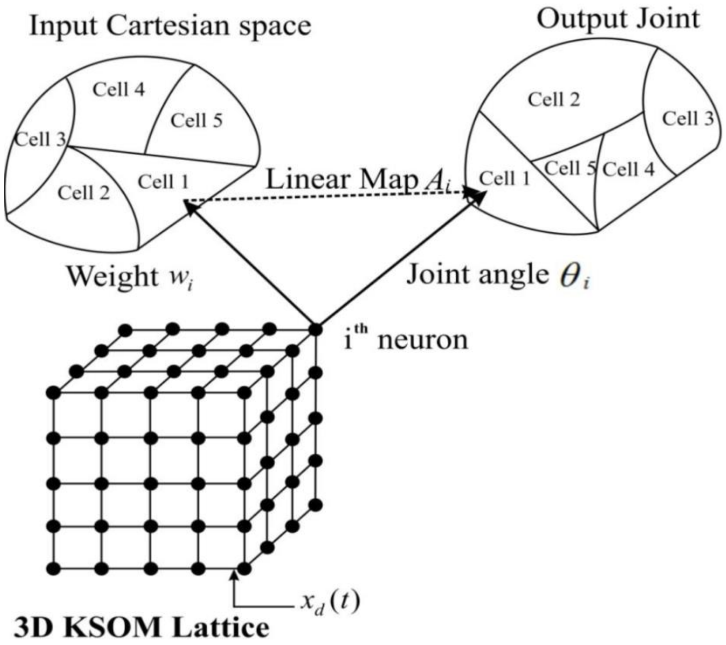
Introduction (cont'd)

- **Forward kinematics data** is collected for the random joint angle configurations – the neural network is trained in **supervisory mode** to learn either the forward map or the inverse map. However, for the redundant robot, the inverse kinematic relationship **cannot be learned** directly with the data, since, it is **one-to-many** relationship.

- Alternatively, the **inverse kinematic relationship** can be directly learned in **unsupervisory mode**, by actuating the robot with the joint angle configurations generated by the neural network and then, adapting the network for the positions reached by the end-effector.
  -- such an approach resolves redundancy in the learning phase.

KSOM Based Kinematic Control of the Exoskeleton

KSOM Architecture



KSOM Based Kinematic Control of the Exoskeleton

KSOM Learning Algorithm

I. Computation of winner neuron

$$\mu = \min_i \|x_d - w_i\|$$

II. Coarse movement

$$\theta_0 = s^{-1} \sum_{i=1}^{N} h_i \{\theta_i + A_i(x_d - w_i)\}$$

III. Fine movement

$$\theta_1 = \theta_0 + s^{-1} \sum_{i=1}^{N} h_i A_i(x_d - x_0)$$

KSOM Based Kinematic Control of the Exoskeleton

KSOM Learning Algorithm

Network Parameter Updation

$$A_i^{new} = A_i^{old} + s^{-1}\alpha h_i \Delta A_i \qquad \Delta A_i = \frac{(\Delta\theta_{01} - A_i \Delta x_{01})\Delta x_{01}^T}{\|\Delta x_{01}\|^2}$$

$$w_i^{new} = w_i^{old} + s^{-1}\alpha h_i \Delta w_i \qquad \Delta w_i = x_d - w_i$$

$$\theta_i^{new} = \theta_i^{old} + s^{-1}\alpha h_i \Delta\theta_i \qquad \Delta\theta_i = \theta_0 - \theta_i - A_i^{new}(x_0 - w_i)$$

KSOM Based Kinematic Control of the Exoskeleton

KSOM Learning Algorithm

Inverse Jacobian Estimation

The fine movement can be rewritten as,

$$\theta_1 - \theta_0 = s^{-1} \sum_{i=1}^{N} h_i \, A_i (x_d - x_0)$$

$$\Delta\theta = s^{-1} \sum_{i=1}^{N} h_i \, A_i (x_d - x_0)$$

$$\dot{\theta} = s^{-1} \sum_{i=1}^{N} h_i \, A_i \dot{x}$$

$$J^{+} \approx s^{-1} \sum_{i=1}^{N} h_i \, A_i$$

Inverse Kinematics of Three Finger Exoskeleton

$$\mathbf{x}_k = \mathbf{f}(\boldsymbol{\theta}_k)$$

$$\dot{\mathbf{x}}_k = \mathbf{J}_k \dot{\boldsymbol{\theta}}_k$$

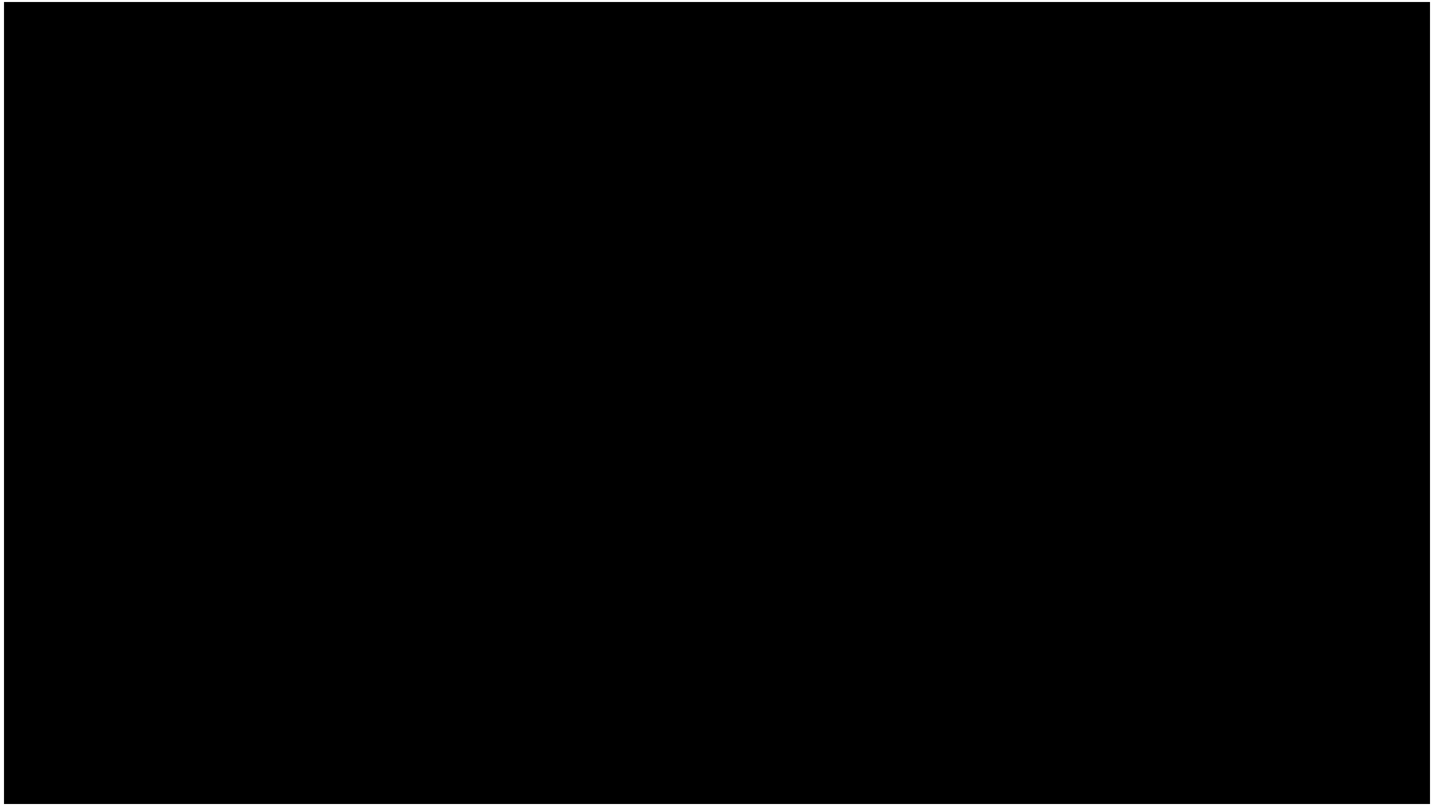
$$\mathbf{J}_k = \frac{\partial \mathbf{f}(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}$$

$$\dot{\boldsymbol{\theta}}_k = \mathbf{J}_k^{+} \dot{\mathbf{x}}_{kd} + (\mathbf{I} - \mathbf{J}_k^{+} \mathbf{J}_k)\mathbf{N}$$

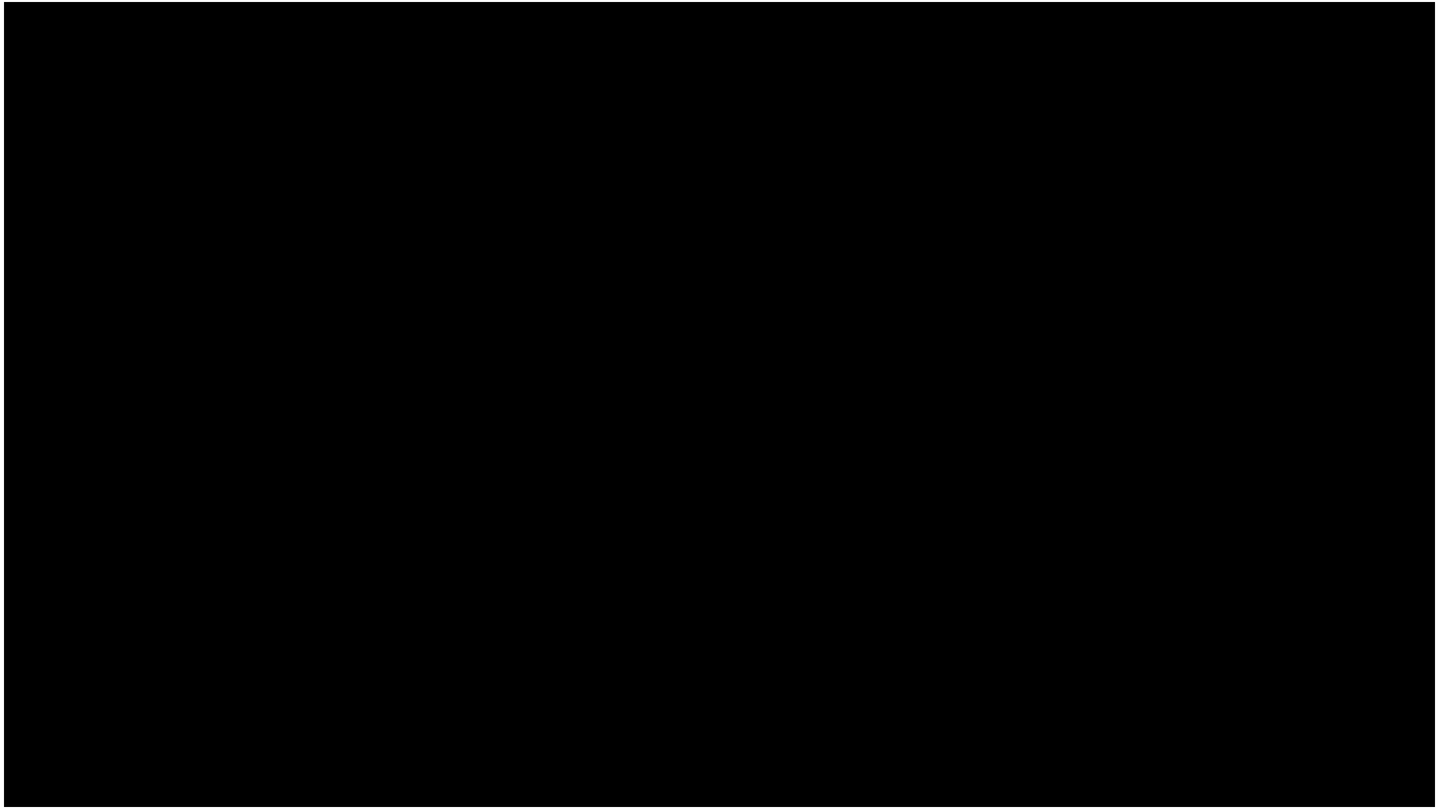
where, $\mathbf{J}_k^{+} = \mathbf{J}_k^{T} (\mathbf{J}_k \mathbf{J}_k^{T})^{-1}$ is the Pseudo Inverse and **N** is an arbitrary vector

$$\dot{\boldsymbol{\theta}}_k = \mathbf{J}_k^{+} \dot{\mathbf{x}}_{kd} + (\mathbf{I} - \mathbf{J}_k^{+} \mathbf{J}_k) k_p \frac{\partial M(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k} \quad ; \quad M(\boldsymbol{\theta}_k) = \sqrt{\det(\mathbf{J}_k \mathbf{J}_k^{T})}$$

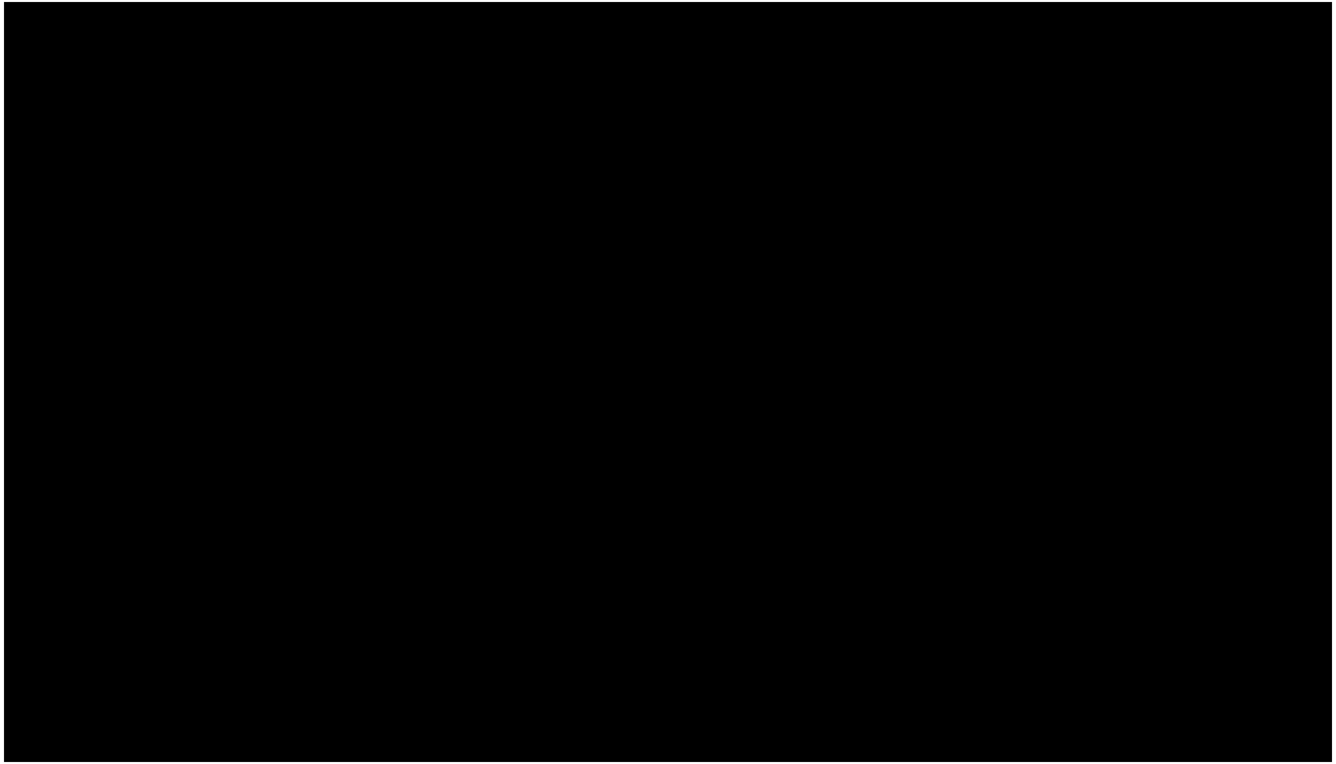
Inverse Kinematics of Three Finger Exoskeleton (cont'd)



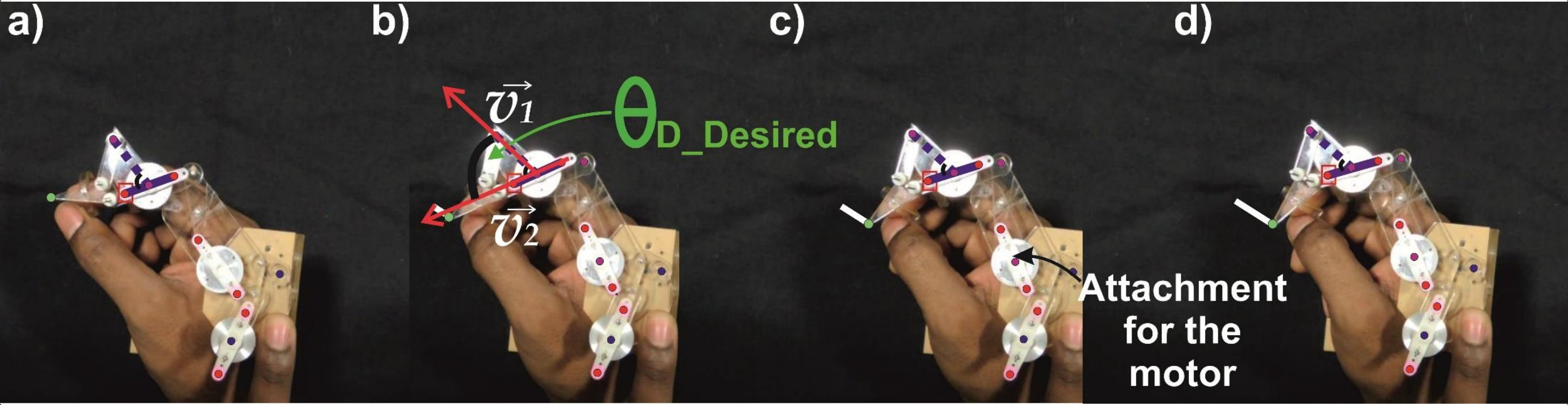
Inverse Kinematics of Three Finger Exoskeleton (cont’d)



Inverse Kinematics of Three Finger Exoskeleton (cont'd)

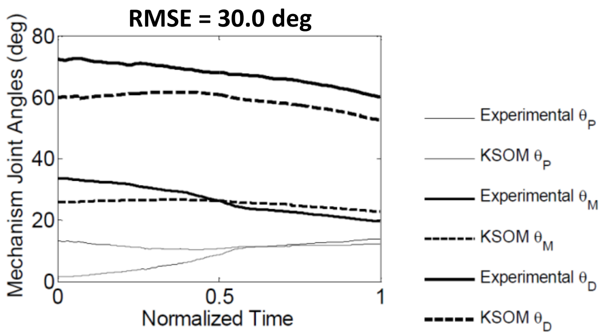


Experimental Angle Computation

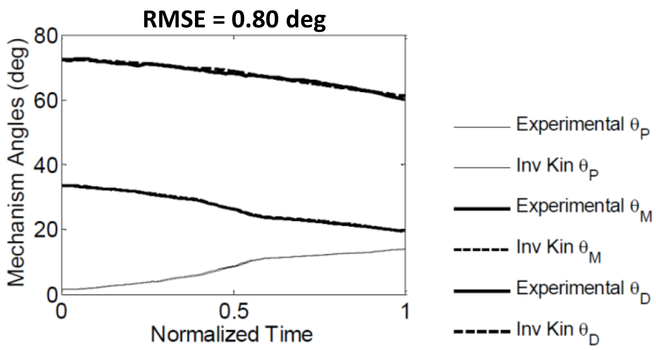


$$\theta_D = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{||\vec{v}_1||\,||\vec{v}_2||}\right)$$

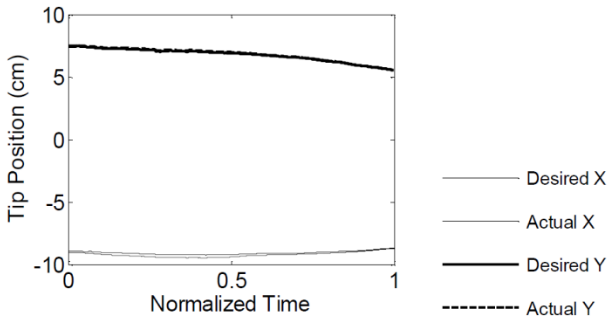
Index Finger Exoskeleton Trajectory Matching



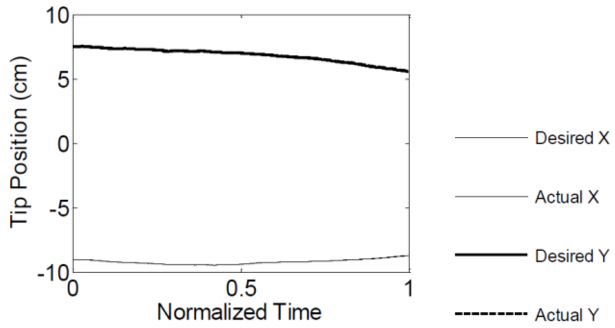
Exoskeleton Joint angle matching (KSOM)



Exoskeleton Joint angle matching (Inv Kin)

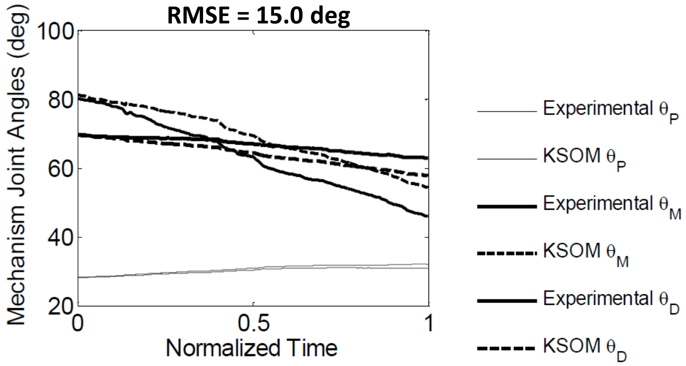


Exoskeleton tip trajectory matching (KSOM)

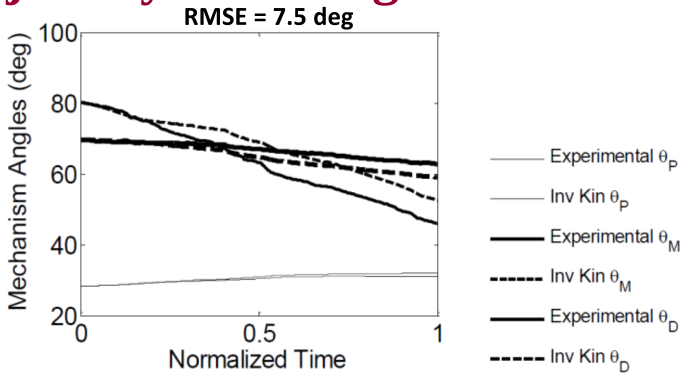


Exoskeleton tip trajectory matching (Inv Kin)

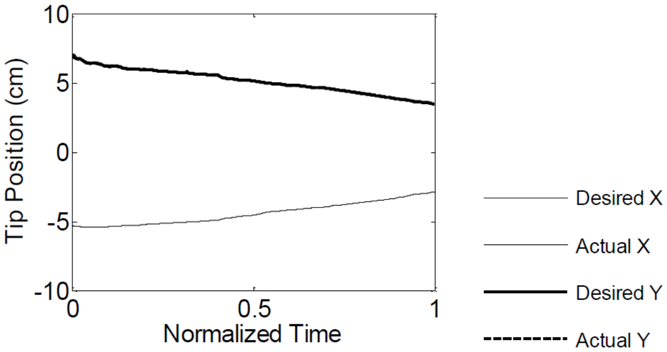
Middle Finger Exoskeleton Trajectory Matching



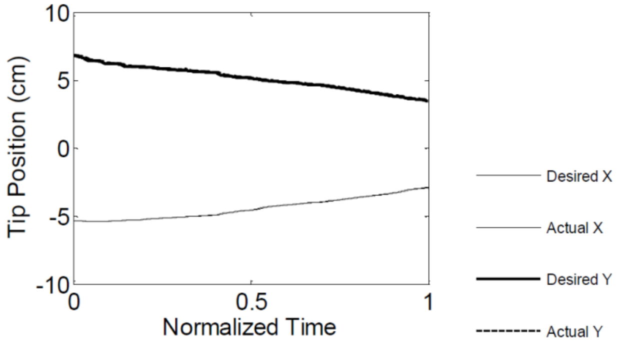
Exoskeleton Joint angle matching (KSOM)



Exoskeleton Joint angle matching (Inv Kin)

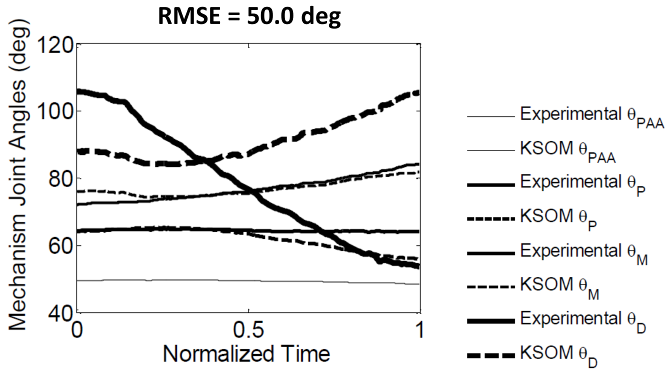


Exoskeleton tip trajectory matching (KSOM)

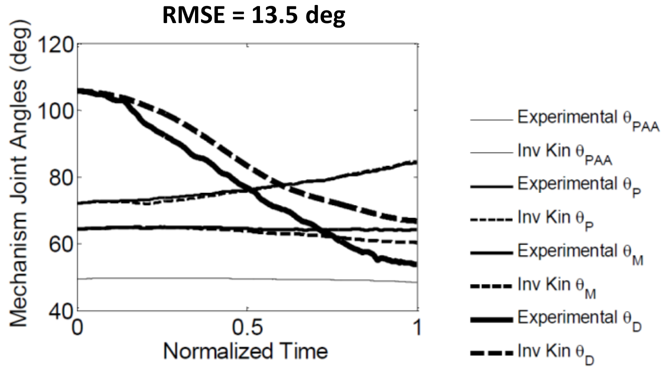


Exoskeleton tip trajectory matching (Inv Kin)

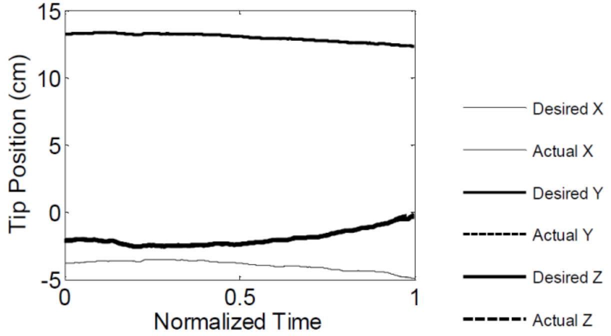
Thumb Exoskeleton Trajectory Matching



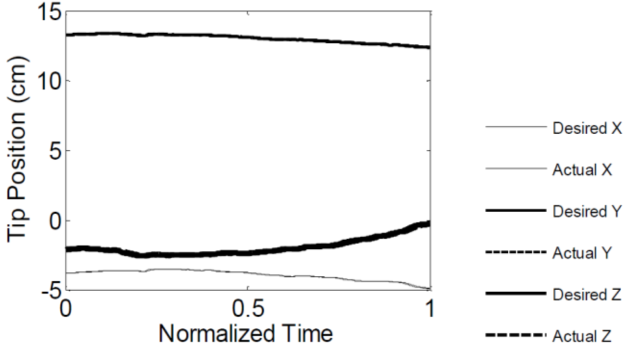
Exoskeleton Joint angle matching (KSOM)



Exoskeleton Joint angle matching (Inv Kin)

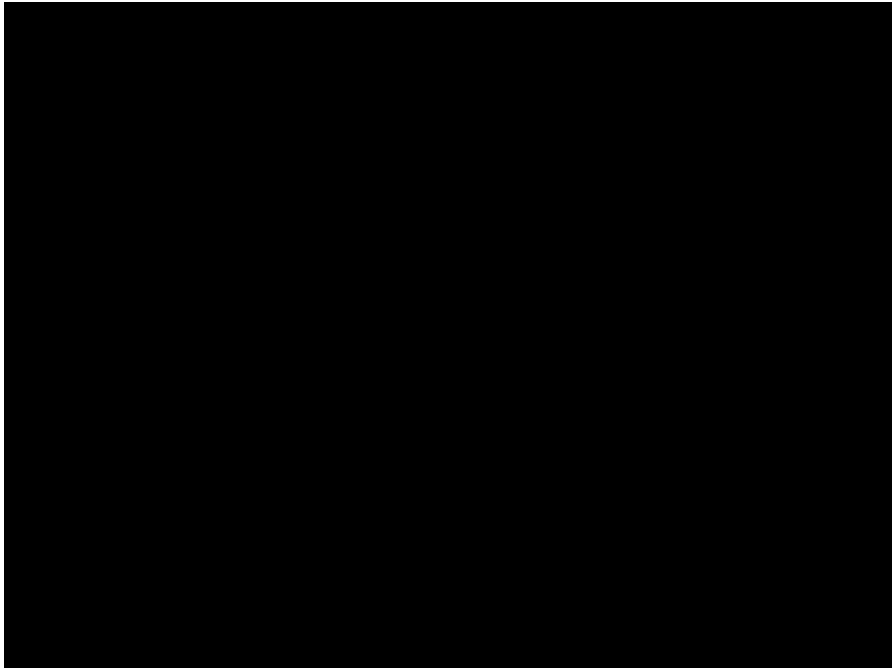


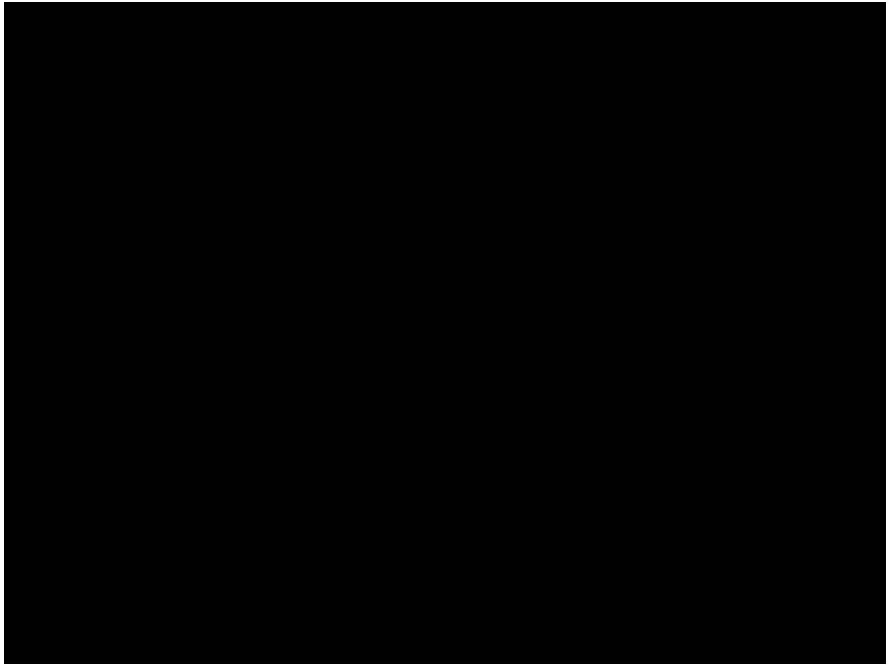
Exoskeleton tip trajectory matching (KSOM)



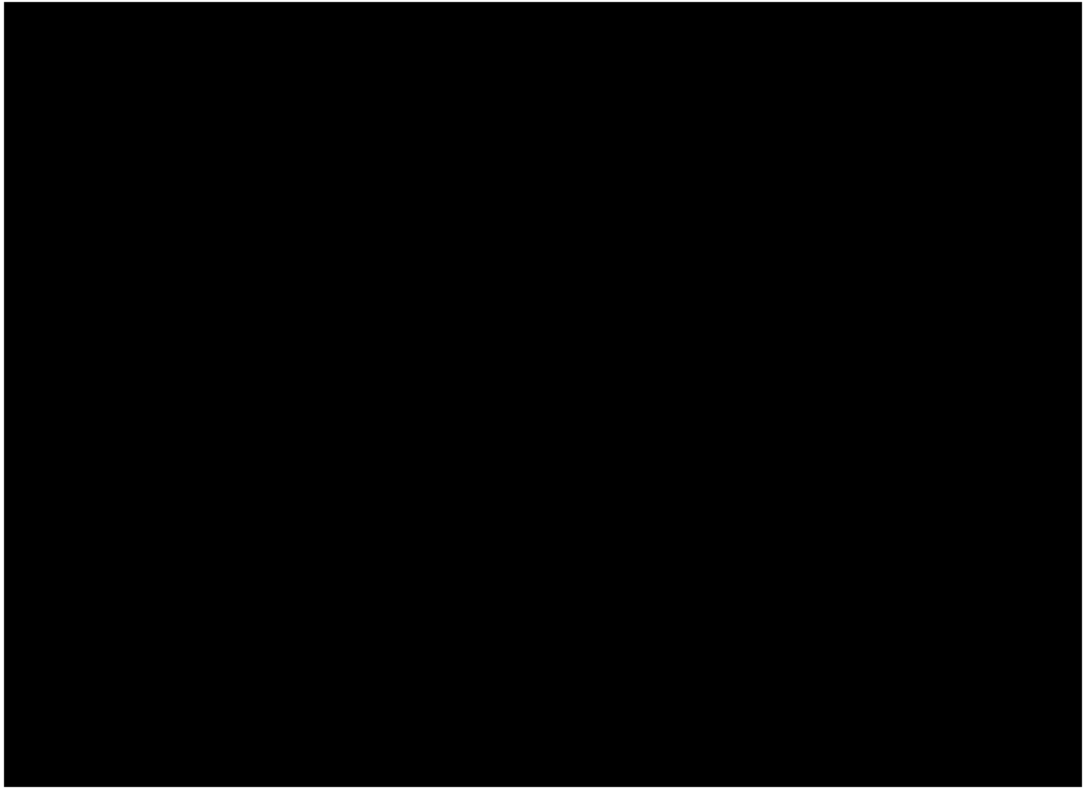
Exoskeleton tip trajectory matching (Inv Kin)

Working Demo

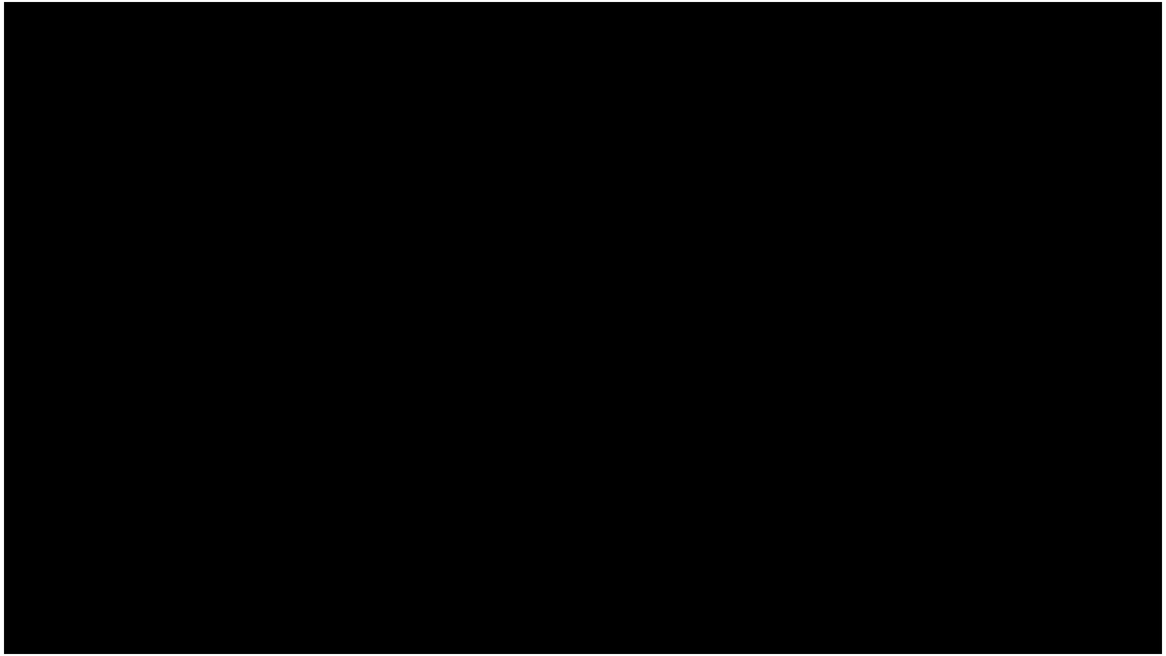




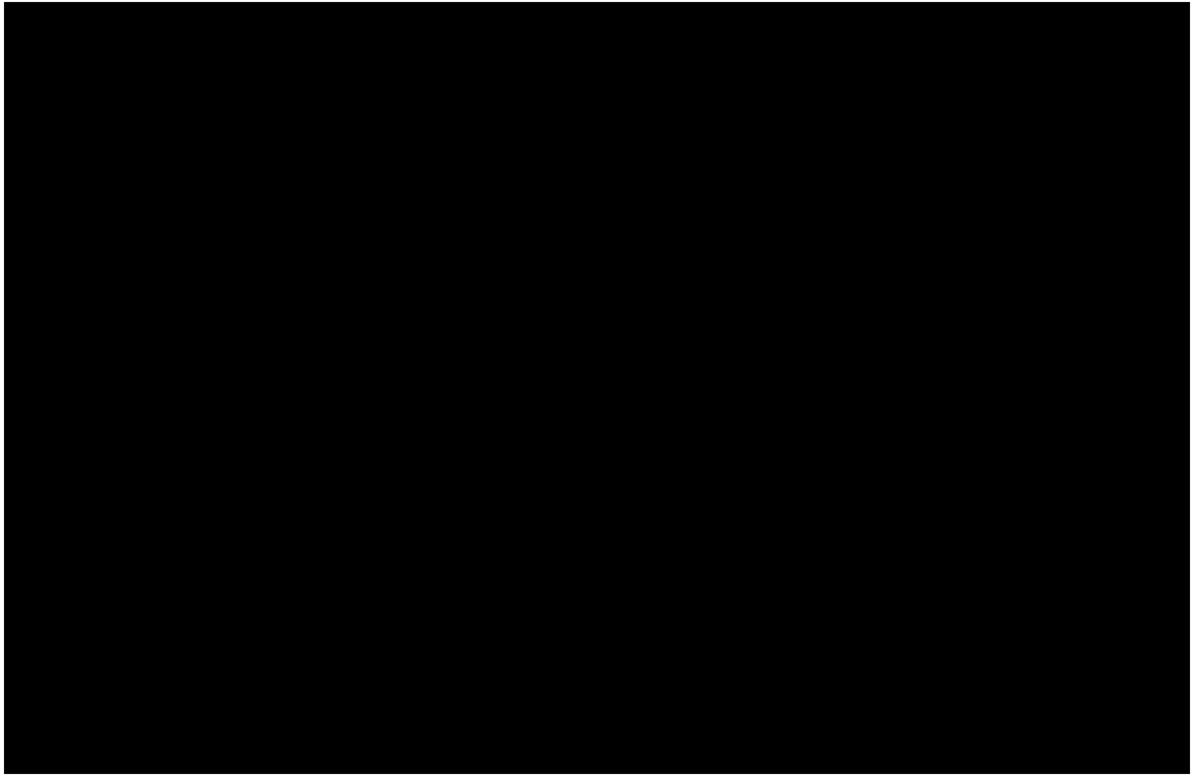
Working Demo



Working Demo



Working Demo



Summary

- The given tip trajectory tracking task performance of the designed exoskeleton is compared between KSOM based scheme and the Inverse kinematics model based scheme.

- From the results it is shown that the Inverse kinematic model tracks the given tip trajectory better than the KSOM based scheme.

References

1. M. Felix Orlando, P. Prem Kumar, Laxmidhar Behera, “Redundancy Resolution of an Index Finger Exoskeleton using Self Organizing Map”, IEEE-AIM (Advanced Intelligent Mechatronics) 2018, Auckland, New Zealand, 9-12 July 2018, pp. 863-868.

2. Laxmidhar Behera and Indrani Kar, Intelligent Systems and Control Principles and Applications, Oxford, U. K.: Oxford Univ. Press, 2009.

3. P. Prem Kumar and Laximdhar Behera, “Visual servoing of redundant manipulator with Jacobian matrix estimation using self-organizing map,” Robotics and Autonomous System, vol. 58(3), pp. 978-990, 2010.

Thank You!