Mobile Manipulator Robot Simulator

A Simulator for Design, Analysis, Redundancy Resolution, and Control of Differential-Drive Wheeled Mobile Manipulator Robots (DDWMMRs)

Tanmay Vilas Samak

PhD Candidate

ARMLab, CU-ICAR

Email: tsamak@clemson.edu

Chinmay Vilas Samak

PhD Candidate

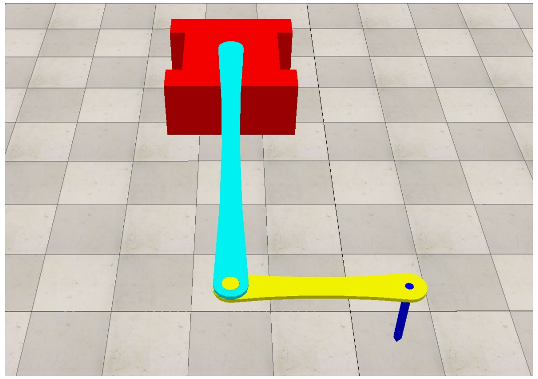
ARMLab, CU-ICAR

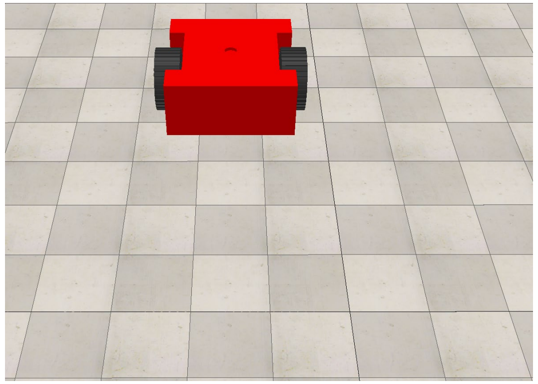
Email: csamak@clemson.edu

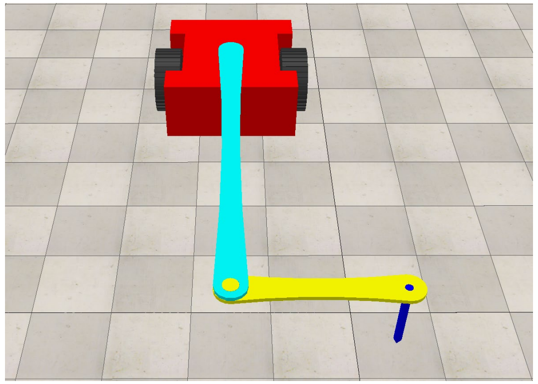
Motivation

Mobile & Manipulator Robots

- Manipulator robots
  - Limited static workspace
  - High accuracy and precision (low uncertainty)
  - Human intervention / separate automation required beyond workspace
- Mobile robots
  - Theoretically infinite (planar) dynamic workspace
  - Moderate accuracy and precision (higher uncertainty)
  - Human intervention / separate automation required for manipulation
- Mobile-manipulator robots
  - Combined benefits of mobile & manipulator robots
  - Coordinated motion control







Objectives & Deliverables

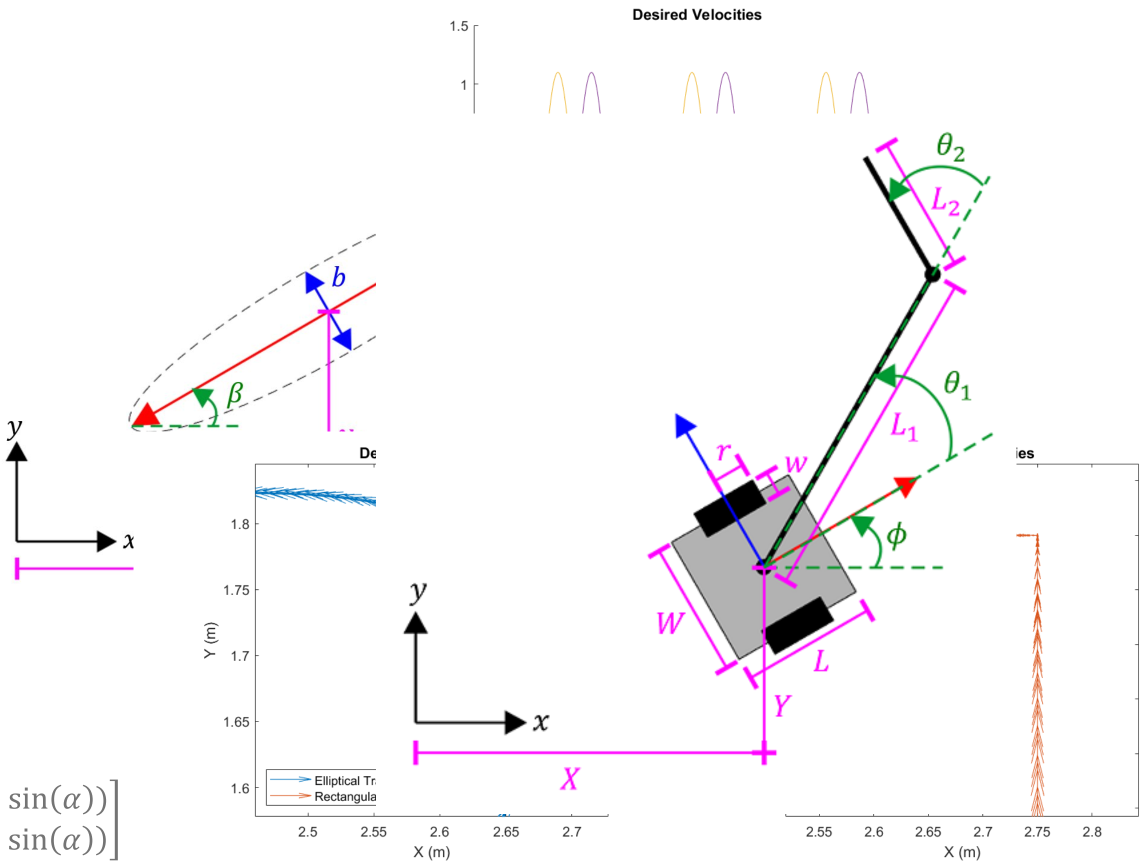
Design, Analysis, Control & Simulation of DDWMMR

- Phase 1: Design and formulation
  - Desired trajectories
  - Mobile-manipulator robot
- Phase 2: Redundancy resolution
  - Pseudoinverse method
  - Augmented task-space method
  - Artificial potential method
- Phase 3: Closed-loop resolved-rate motion control
  - Configuration-space control
  - Task-space control
- Phase 4: MATLAB GUI design
- Phase 5: CoppeliaSim-MATLAB co-simulation setup

Solution Approach

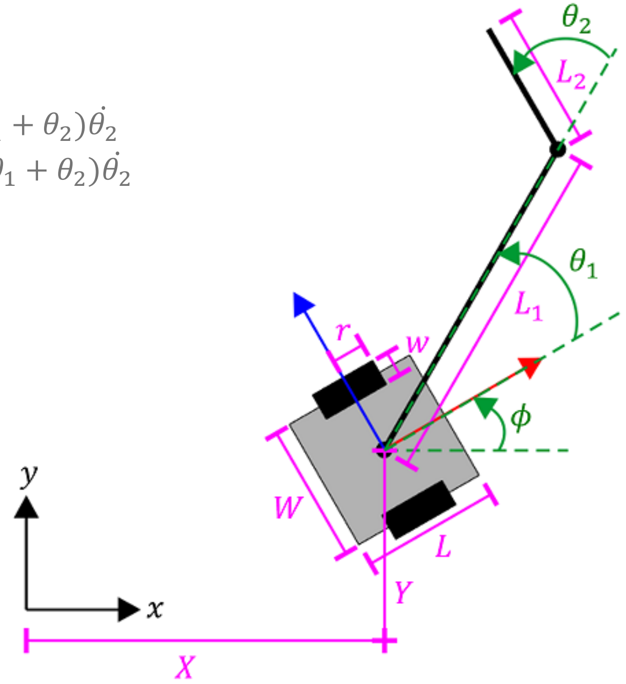
Design

- Desired trajectories
  - $\begin{bmatrix} x_E \\ y_E \end{bmatrix}_d = f(x_0, y_0, a, b, \beta, N, T)$
  - Elliptical trajectory
    - $\begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} a\cos(\alpha) \\ b\sin(\alpha) \end{bmatrix}$
    - $\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} -a\sin(\alpha)\dot{\alpha} \\ b\cos(\alpha)\,\dot{\alpha} \end{bmatrix}$
  - Rectangular trajectory
    - $\begin{bmatrix} x_E \\ y_E \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} a(|\cos(\alpha)|\cos(\alpha) - |\sin(\alpha)|\sin(\alpha)) \\ b(|\cos(\alpha)|\cos(\alpha) + |\sin(\alpha)|\sin(\alpha)) \end{bmatrix}$
    - $\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} a\left(-\frac{\sin(\alpha)*\cos(\alpha)}{|\cos(\alpha)|} * \cos(\alpha) * \dot{\alpha} - |\cos(\alpha)| * \sin(\alpha) * \dot{\alpha} - \frac{\sin(\alpha)*\cos(\alpha)}{|\sin(\alpha)|} * \sin(\alpha) * \dot{\alpha} - |\sin(\alpha)| * \cos(\alpha) * \dot{\alpha}\right) \\ a\left(-\frac{\sin(\alpha)*\cos(\alpha)}{|\cos(\alpha)|} * \cos(\alpha) * \dot{\alpha} - |\cos(\alpha)| * \sin(\alpha) * \dot{\alpha} + \frac{\sin(\alpha)*\cos(\alpha)}{|\sin(\alpha)|} * \sin(\alpha) * \dot{\alpha} + |\sin(\alpha)| * \cos(\alpha) * \dot{\alpha}\right) \end{bmatrix}$
- Mobile-manipulator robot



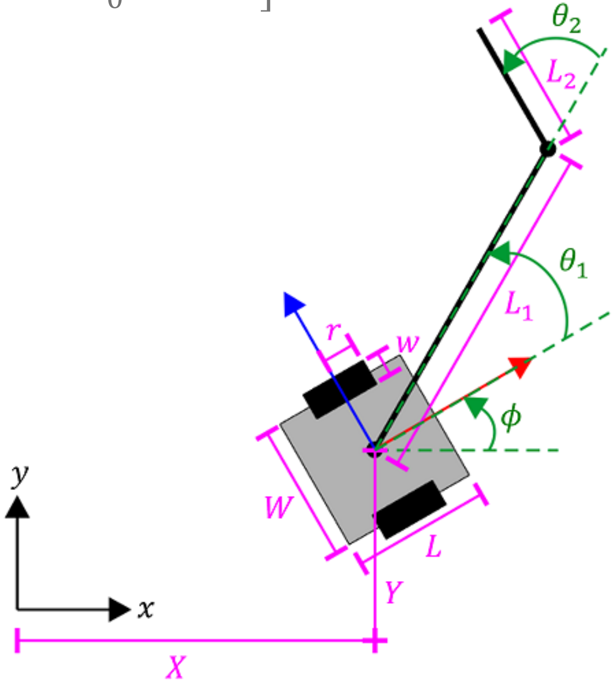
Formulation

- Forward kinematics:
  - $$\begin{cases} x_E = x + L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) \\ y_E = y + L_1\sin(\phi+\theta_1) + L_2\sin(\phi+\theta_1+\theta_2) \\ \phi_E = \phi + \theta_1 + \theta_2 \end{cases}$$
- Differential kinematics:
  - $$\begin{cases} \dot{x_E} = \dot{x} - L_1\sin(\phi+\theta_1)\dot{\phi} - L_1\sin(\phi+\theta_1)\dot{\theta_1} - L_2\sin(\phi+\theta_1+\theta_2)\dot{\phi} - L_2\sin(\phi+\theta_1+\theta_2)\dot{\theta_1} - L_2\sin(\phi+\theta_1+\theta_2)\dot{\theta_2} \\ \dot{y_E} = \dot{y} + L_1\cos(\phi+\theta_1)\dot{\phi} + L_1\cos(\phi+\theta_1)\dot{\theta_1} + L_2\cos(\phi+\theta_1+\theta_2)\dot{\phi} + L_2\cos(\phi+\theta_1+\theta_2)\dot{\theta_1} + L_2\cos(\phi+\theta_1+\theta_2)\dot{\theta_2} \\ \dot{\phi_E} = \dot{\phi} + \dot{\theta_1} + \dot{\theta_2} \end{cases}$$
- Non-holonomic constraint:
  - $-\dot{x}\sin(\phi) + \dot{y}\cos(\phi) = 0$



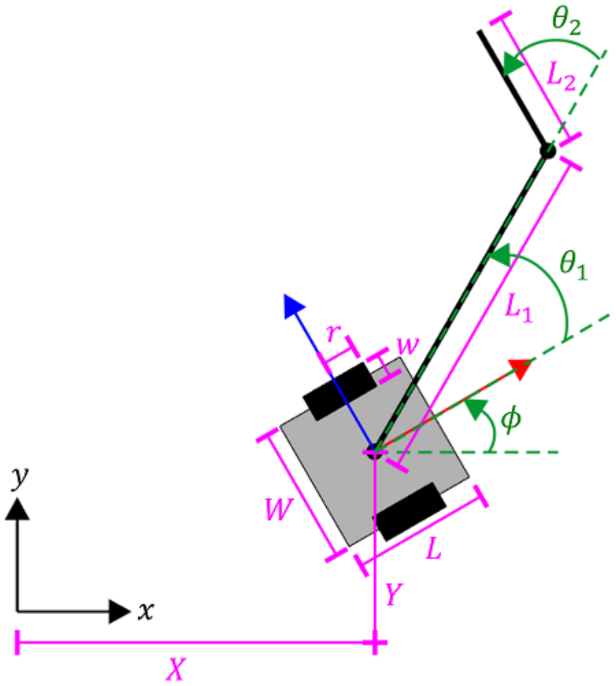
Formulation

- Jacobian matrix:
  - $$J(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \end{bmatrix}$$
- Task-space variables:
  - $$\boldsymbol{X} = \begin{bmatrix} x_E \\ y_E \\ \phi_E \\ 0 \end{bmatrix}, \boldsymbol{X_d} = \begin{bmatrix} x_{E\,d} \\ y_{E\,d} \\ \phi_{E\,d} \\ 0 \end{bmatrix}, \dot{\boldsymbol{X}} = \begin{bmatrix} \dot{x_E} \\ \dot{y_E} \\ \dot{\phi_E} \\ 0 \end{bmatrix}, \dot{\boldsymbol{X_d}} = \begin{bmatrix} \dot{x_E}_d \\ \dot{y_E}_d \\ \dot{\phi_E}_d \\ 0 \end{bmatrix}$$
- Configuration-space variables :
  - $$\boldsymbol{q} = \begin{bmatrix} x \\ y \\ \phi \\ \theta_1 \\ \theta_2 \end{bmatrix}, \boldsymbol{q_d} = \begin{bmatrix} x_d \\ y_d \\ \phi_d \\ \theta_{1d} \\ \theta_{2d} \end{bmatrix}, \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta_1} \\ \dot{\theta_2} \end{bmatrix}, \dot{\boldsymbol{q_d}} = \begin{bmatrix} \dot{x_d} \\ \dot{y_d} \\ \dot{\phi_d} \\ \dot{\theta_1}_d \\ \dot{\theta_2}_d \end{bmatrix}$$



Formulation

- Wheel velocities:
  - $$\begin{cases} v = \dot{x}\cos(\phi) + \dot{y}\sin(\phi) = \frac{(\dot{\phi}_l + \dot{\phi}_r)}{2r} \\ \omega = \dot{\phi} = \frac{(\dot{\phi}_r - \dot{\phi}_l)}{Lr} \end{cases}$$
  - $$\begin{cases} \dot{\phi}_l = \frac{2[\dot{x}\cos(\phi) + \dot{y}\sin(\phi)] - L\dot{\phi}}{2r} \\ \dot{\phi}_r = \frac{2[\dot{x}\cos(\phi) + \dot{y}\sin(\phi)] + L\dot{\phi}}{2r} \end{cases}$$



Redundancy Resolution

- Pseudoinverse method
  - $$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \end{bmatrix}$$
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{\#}\dot{\boldsymbol{X}}$
- Augmented task-space method
  - $$\dot{\theta}_1 + \dot{\theta}_2 = 0 \quad \Rightarrow \quad \boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$
  - $$\dot{\theta}_1 = 0 \quad \Rightarrow \quad \boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{X}}$

Redundancy Resolution

- Artificial potential method $\mathrm{V} = 0.66\phi^2 + {\theta_1}^2 + 0.25{\theta_2}^2$
  - $V = K_1\theta_1^2 + K_2\theta_2^2 + K_3\phi^2$
  - $-\nabla V = \begin{bmatrix} -\frac{\partial V}{\partial \theta_1} \\ -\frac{\partial V}{\partial \theta_2} \\ -\frac{\partial V}{\partial \phi} \end{bmatrix} = \begin{bmatrix} -2K_1\theta_1 \\ -2K_2\theta_2 \\ -2K_3\phi \end{bmatrix}$
  - $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{\#}\dot{\boldsymbol{X}} + \underbrace{[\boldsymbol{I} - \boldsymbol{J}(\boldsymbol{q})^{\#}\boldsymbol{J}(\boldsymbol{q})]}_{\text{Null space filter}}\boldsymbol{z} = \boldsymbol{J}(\boldsymbol{q})^{\#}\dot{\boldsymbol{X}} + [\boldsymbol{I} - \boldsymbol{J}(\boldsymbol{q})^{\#}\boldsymbol{J}(\boldsymbol{q})](-\nabla \boldsymbol{V})$

Closed-Loop Resolved Rate Motion Control (Config Space)

- $\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

- $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}(\boldsymbol{q}_{\boldsymbol{d}} - \boldsymbol{q})$, where $\dot{\boldsymbol{q}}_{\boldsymbol{d}} = \boldsymbol{J}(\boldsymbol{q})^{-1}\dot{\boldsymbol{X}}_{\boldsymbol{d}}$ and $\boldsymbol{K} = \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 \\ 0 & 0 & K_3 & 0 & 0 \\ 0 & 0 & 0 & K_4 & 0 \\ 0 & 0 & 0 & 0 & K_5 \end{bmatrix}$ with $\underbrace{K_i}_{\text{Negative pole}} = \frac{1}{\underbrace{\tau_i}_{\text{Time constant}}}$

$$\Rightarrow \underbrace{(\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \dot{\boldsymbol{q}})}_{\dot{\boldsymbol{q}}_{\boldsymbol{e}}} + \boldsymbol{K}\underbrace{(\boldsymbol{q}_{\boldsymbol{d}} - \boldsymbol{q})}_{\boldsymbol{q}_{\boldsymbol{e}}} = 0$$

$$\Rightarrow \dot{\boldsymbol{q}}_{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{q}_{\boldsymbol{e}} = 0$$

$$\Rightarrow \underbrace{\boldsymbol{q}_{\boldsymbol{e}}(t)}_{\text{Error at 't'}} = \underbrace{\boldsymbol{q}_{\boldsymbol{e}}(0)}_{\text{Initial error}} \underbrace{e^{-\boldsymbol{K}t}}_{\text{Decay with 't'}} \Rightarrow \begin{cases} \boldsymbol{q}_{\boldsymbol{e}}(t) = \boldsymbol{q}_{\boldsymbol{e}}(0), & t = 0 \\ \boldsymbol{q}_{\boldsymbol{e}}(t) = 0, & t = \infty \end{cases}$$

Closed-Loop Resolved Rate Motion Control (Task Space)

- $$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_1\sin(\phi+\theta_1) - L_2\sin(\phi+\theta_1+\theta_2) & -L_2\sin(\phi+\theta_1+\theta_2) \\ 0 & 1 & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_1\cos(\phi+\theta_1) + L_2\cos(\phi+\theta_1+\theta_2) & L_2\cos(\phi+\theta_1+\theta_2) \\ 0 & 0 & 1 & 1 & 1 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

- $\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})^{-1}[\dot{\boldsymbol{X}}_{\boldsymbol{d}} + \boldsymbol{K}(\boldsymbol{X}_{\boldsymbol{d}} - \boldsymbol{X})]$, where $\boldsymbol{X} = FK(\boldsymbol{q})$ and $\boldsymbol{K} = \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ with $\underbrace{K_i}_{\text{Negative pole}} = \dfrac{1}{\underbrace{\tau_i}_{\text{Time constant}}}$

$$\Rightarrow \dot{\boldsymbol{X}}_{\boldsymbol{d}} + \boldsymbol{K}(\boldsymbol{X}_{\boldsymbol{d}} - \boldsymbol{X}) = \underbrace{\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}}_{\dot{\boldsymbol{X}}}$$

$$\Rightarrow \dot{\boldsymbol{X}}_{\boldsymbol{d}} + \boldsymbol{K}(\boldsymbol{X}_{\boldsymbol{d}} - \boldsymbol{X}) = \dot{\boldsymbol{X}}$$

$$\Rightarrow \underbrace{(\dot{\boldsymbol{X}}_{\boldsymbol{d}} - \dot{\boldsymbol{X}})}_{\dot{\boldsymbol{X}}_e} + \boldsymbol{K}\underbrace{(\boldsymbol{X}_{\boldsymbol{d}} - \boldsymbol{X})}_{\boldsymbol{X}_e} = 0$$

$$\Rightarrow \dot{\boldsymbol{X}}_{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{X}_{\boldsymbol{e}} = 0$$

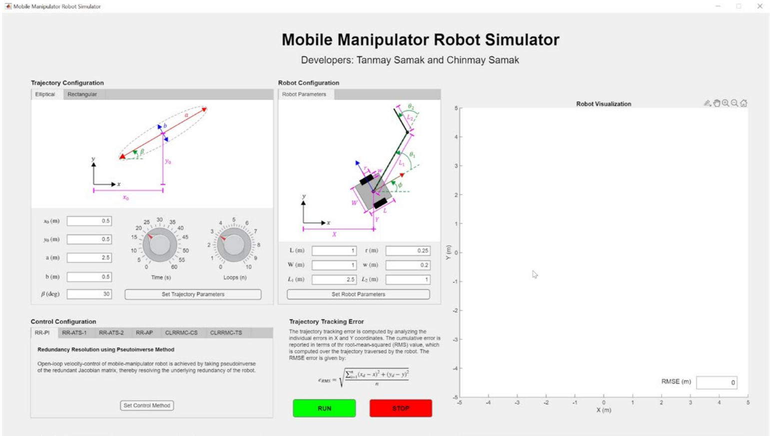
$$\Rightarrow \underbrace{\boldsymbol{X}_{\boldsymbol{e}}(t)}_{\text{Error at 't'}} = \underbrace{\boldsymbol{X}_{\boldsymbol{e}}(0)}_{\text{Initial error}} \underbrace{e^{-\boldsymbol{K}t}}_{\text{Decay with 't'}} \Rightarrow \begin{cases} \boldsymbol{X}_{\boldsymbol{e}}(t) = \boldsymbol{X}_{\boldsymbol{e}}(0), & t = 0 \\ \boldsymbol{X}_{\boldsymbol{e}}(t) = 0, & t = \infty \end{cases}$$

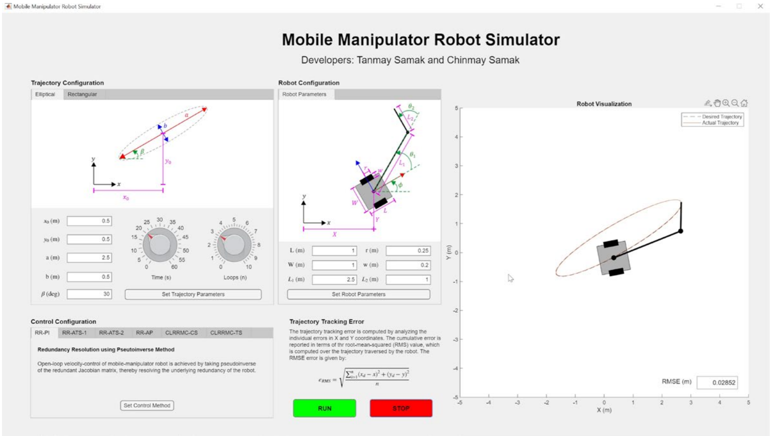
Results

Redundancy Resolution – Pseudoinverse Method

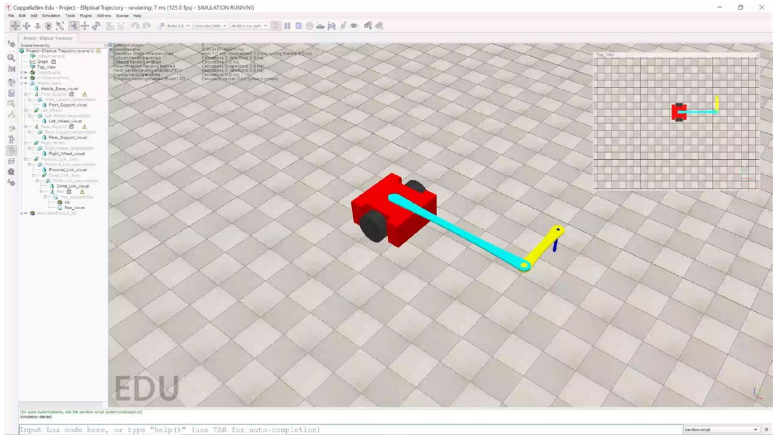
Elliptical Trajectory | Rectangular Trajectory

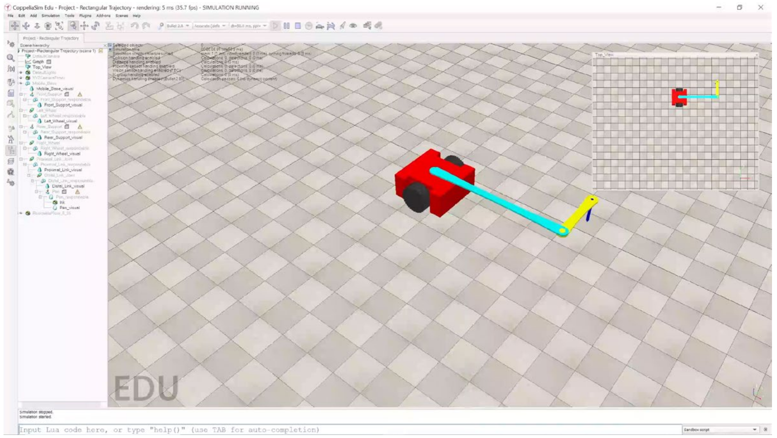
MATLAB-GUI





MATLAB-CoppeliaSim





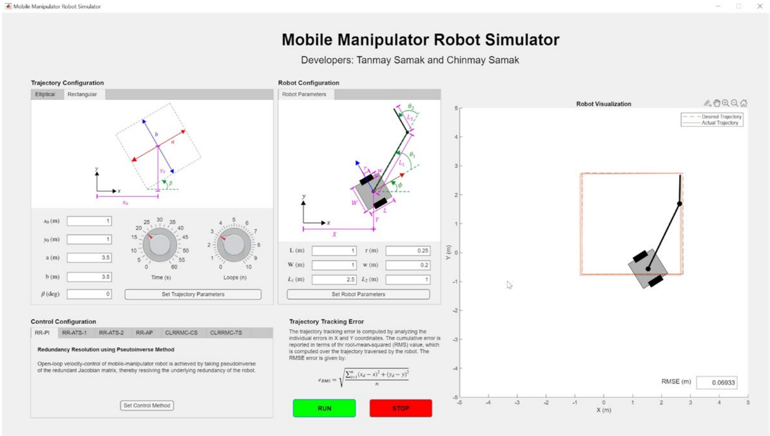
Redundancy Resolution – Augmented Task-Space Method 1

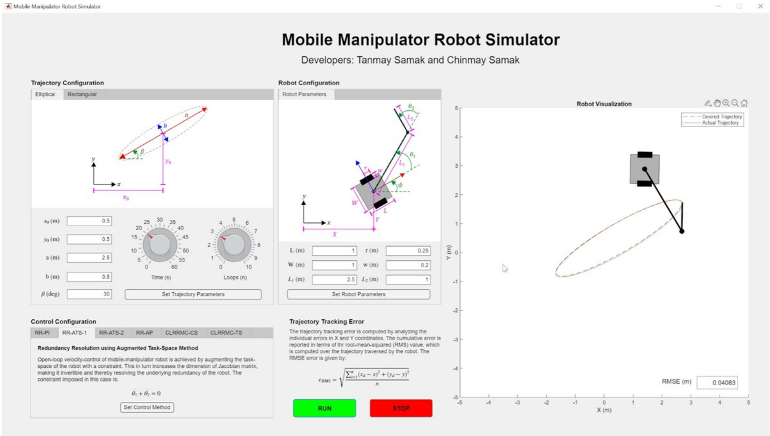
$\dot{\theta}_1 + \dot{\theta}_2 = 0$

Elliptical Trajectory

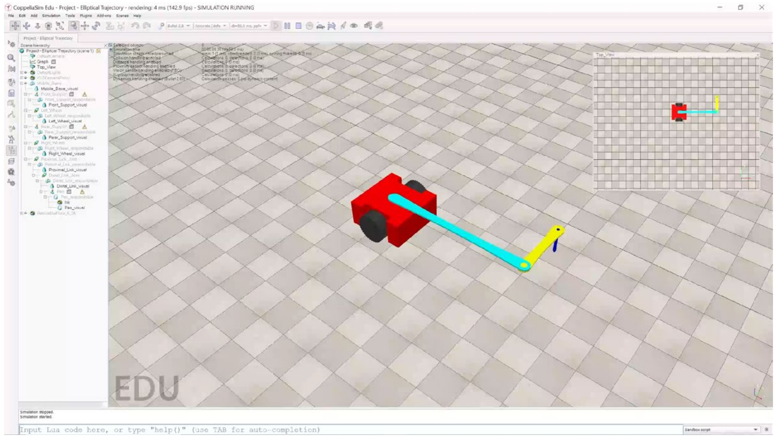
Rectangular Trajectory

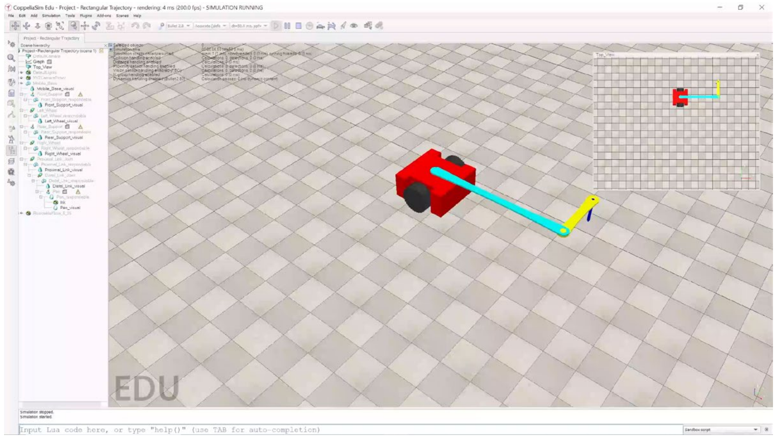
MATLAB-GUI





MATLAB-CoppeliaSim





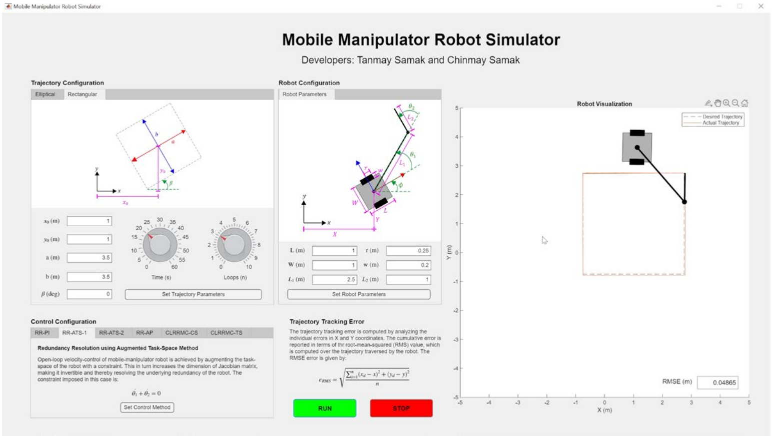
Redundancy Resolution – Augmented Task-Space Method 2

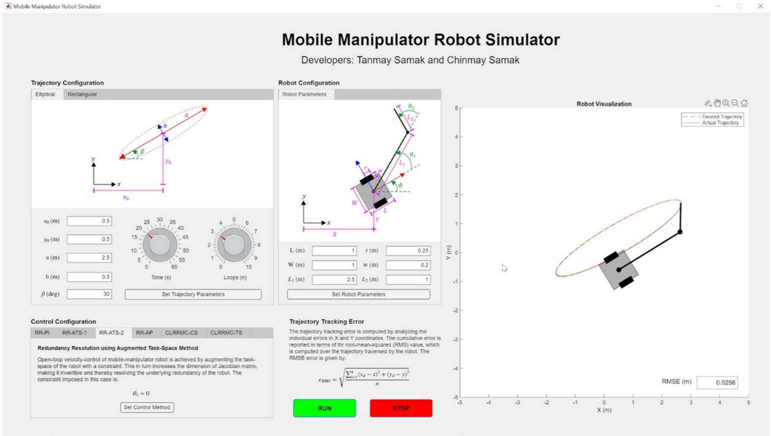
$\dot{\theta}_1 = 0$

Elliptical Trajectory

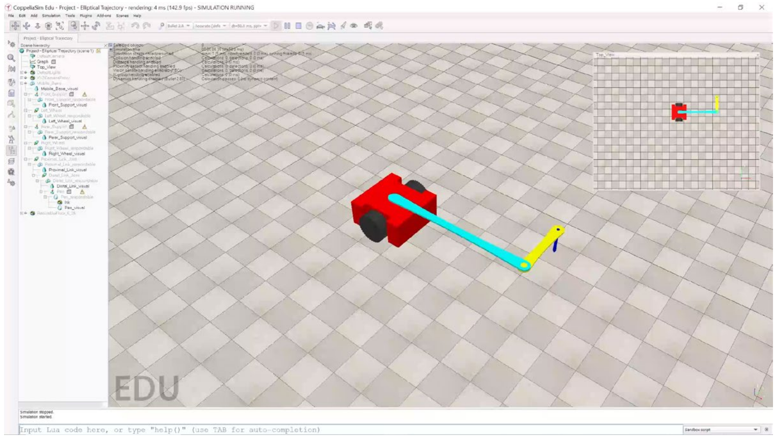
Rectangular Trajectory

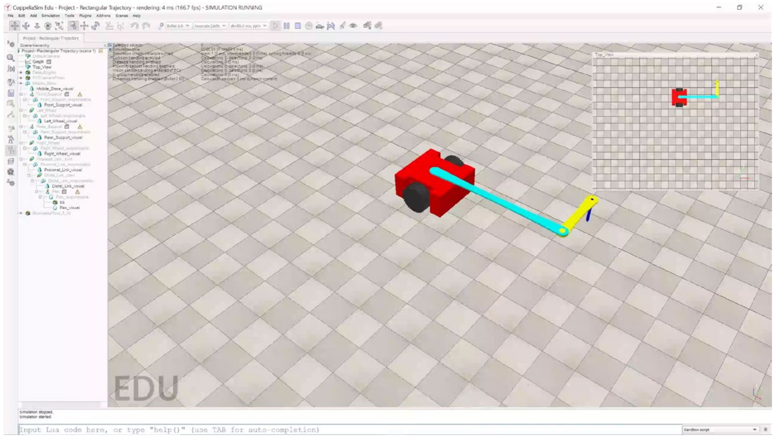
MATLAB-GUI





MATLAB-CoppeliaSim

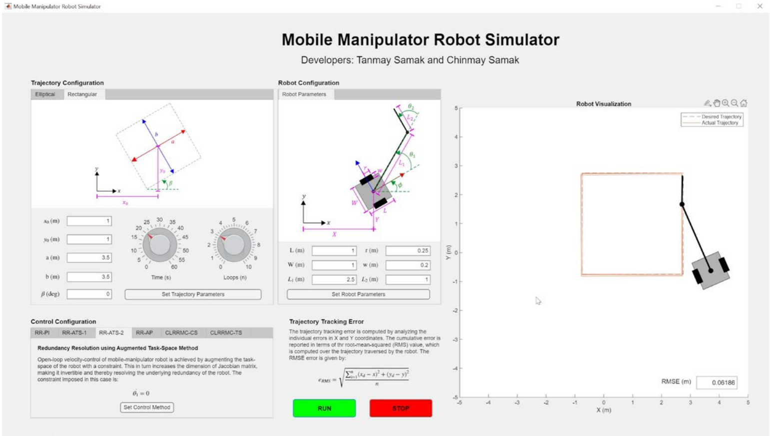


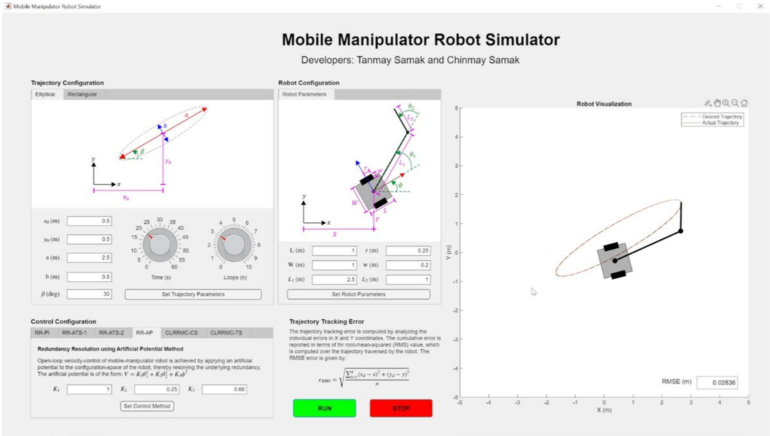


Redundancy Resolution – Artificial Potential Method

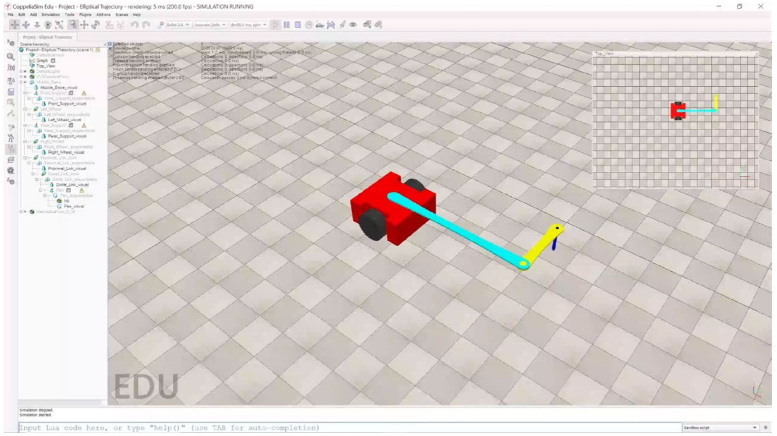
Elliptical Trajectory | Rectangular Trajectory

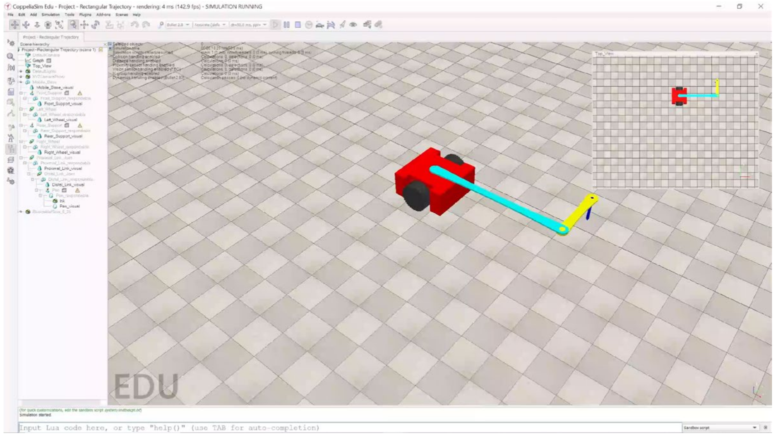
MATLAB-GUI





MATLAB-CoppeliaSim

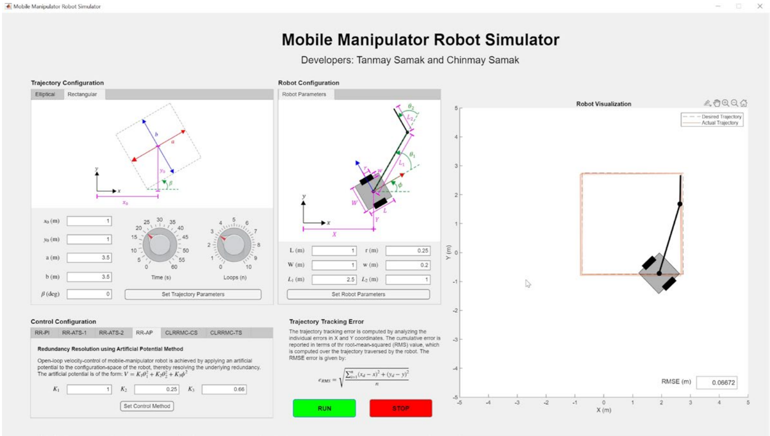


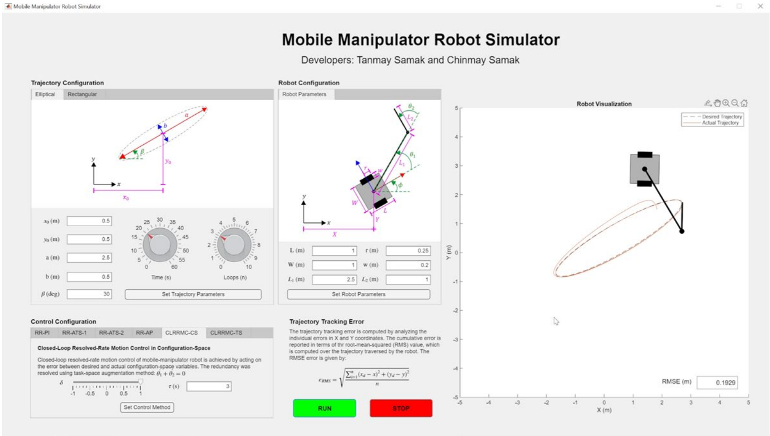


Closed-Loop Resolved Rate Motion Control (Config Space)

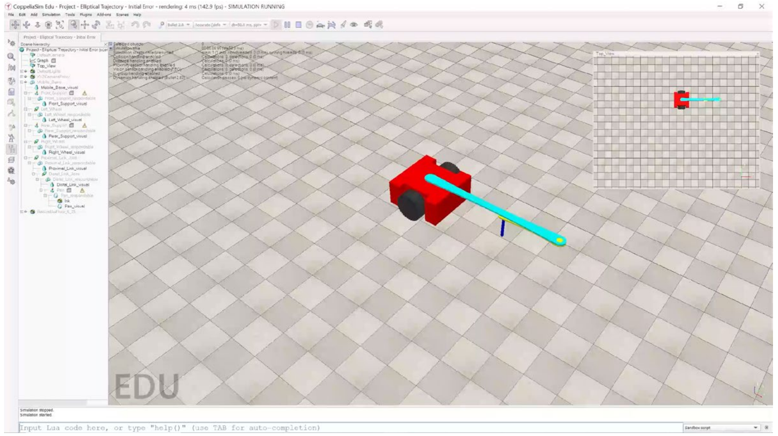
Elliptical Trajectory | Rectangular Trajectory

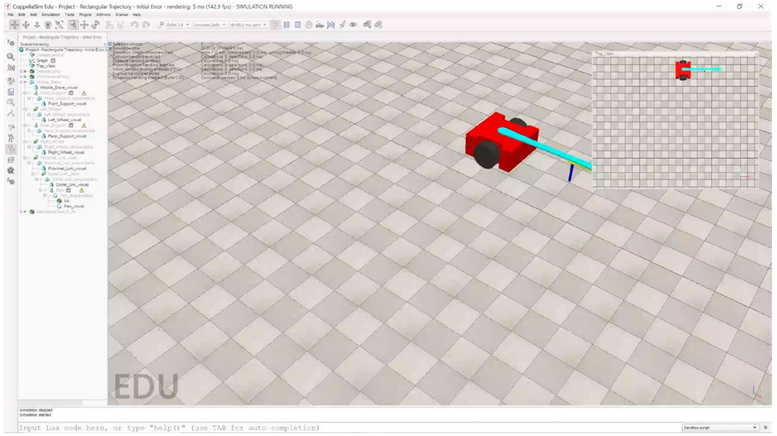
MATLAB-GUI





MATLAB-CoppeliaSim

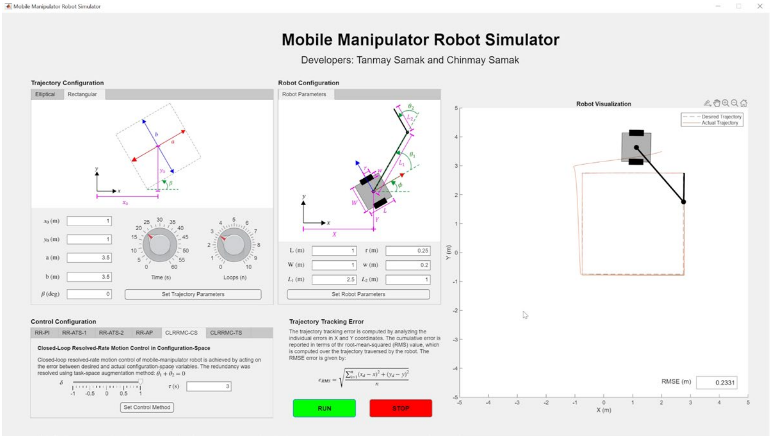


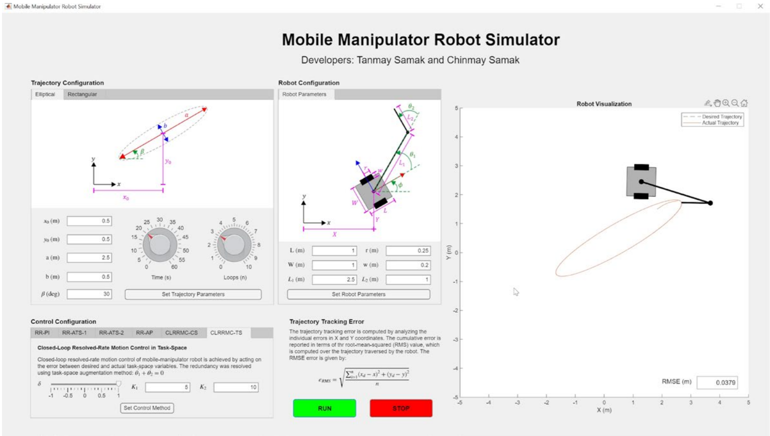


Closed-Loop Resolved Rate Motion Control (Task Space)

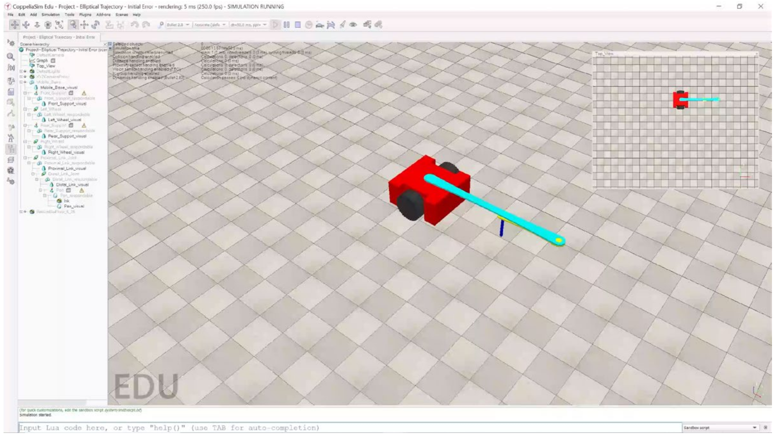
Elliptical Trajectory | Rectangular Trajectory

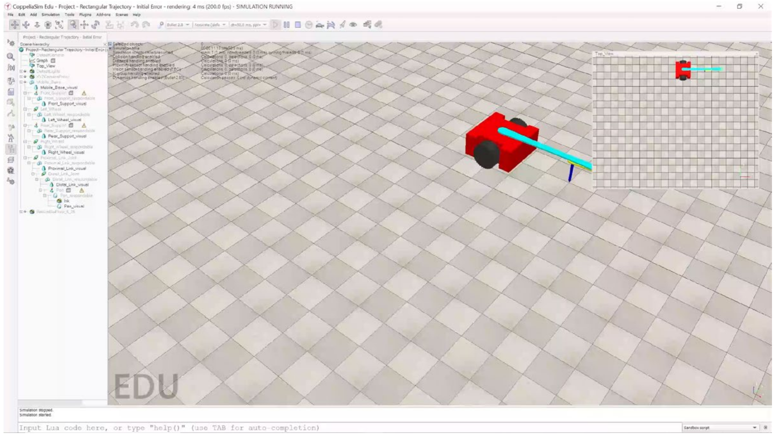
MATLAB-GUI



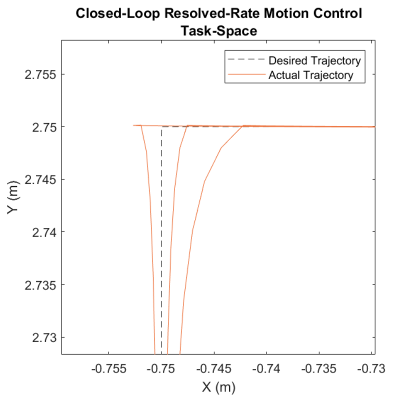


MATLAB-CoppeliaSim





Concluding Remarks



Peculiar Observations & Concluding Remarks

- Initial conditions, robot & trajectory parameters matter a lot!
  - Choose fairly close initial condition with adequate trajectory resolution & robot parameters
- Errors can creep in due to approximations in numerical method & pseudoinverse calculations
  - Choose numerical method wisely (forward Euler is just 1st order fixed-step RK method!)
  - Choose timestep (Δt) wisely
- As time constant (τ) for controller increases, its sluggishness increases (as expected)
- Rectangular trajectory has higher tracking error due to sharp corners (either start turning earlier or pass ahead and then merge back!)
- Dynamic simulation can cause drift in results due to solver used, timestep used, etc.
- CLRRMC-TS $\geq$ CLRRMC-CS $\geq$ RR-AP $\geq$ RR-ATS (depends on choice of auxiliary constraint) $\geq$ RR-PI