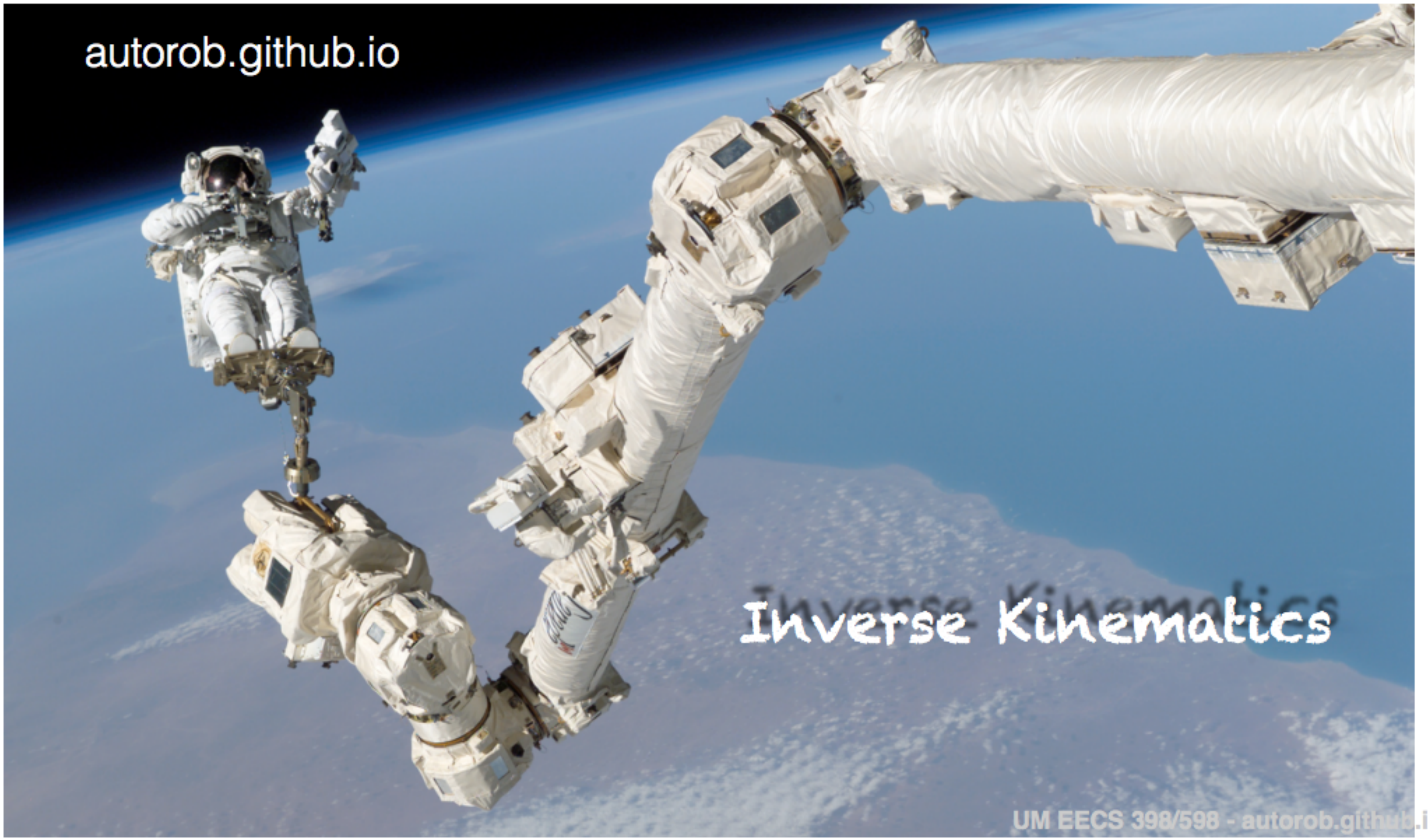
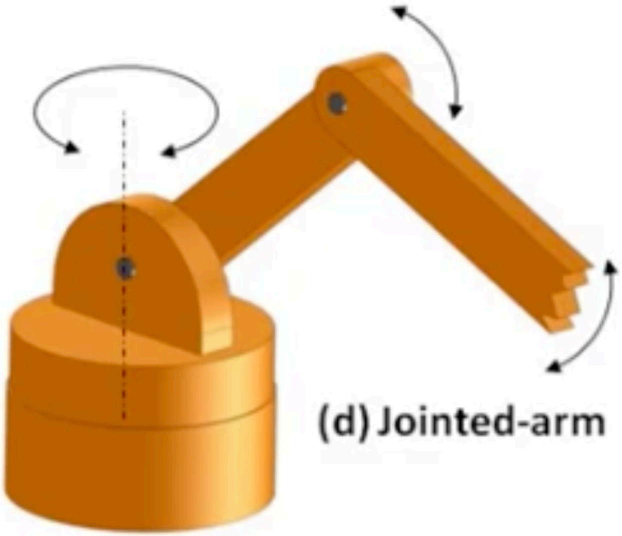
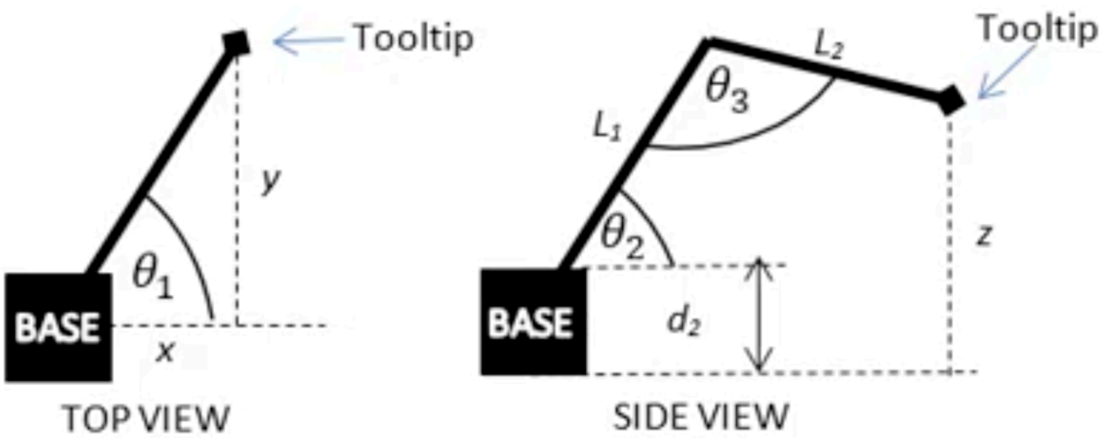
Nonlinear Equations

Nonlinear system of equations



Robotic arms





https://www.youtube.com/watch?v=NRgNDlVtmz0 (Robotic arm 1)

https://www.youtube.com/watch?v=9DqRkLQ5Sv8 (Robotic arm 2)

https://www.youtube.com/watch?v=DZ_ocmY8xEI (Blender)

Inverse Kinematics

Nonlinear system of equations

Goal: Solve $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$ for $\boldsymbol{f}: \mathcal{R}^n \to \mathcal{R}^n$

Newton's method

Approximate the nonlinear function $\boldsymbol{f(x)}$ by a linear function using Taylor expansion:

Newton's method

Algorithm:

Convergence:

- Typically has quadratic convergence
- Drawback: Still only locally convergent

Cost:

- Main cost associated with computing the Jacobian matrix and solving the Newton step.

Example

Consider solving the nonlinear system of equations

$$2 = 2y + x$$
$$4 = x^2 + 4y^2$$

What is the result of applying one iteration of Newton's method with the following initial guess?

$$\boldsymbol{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Newton's method

$\boldsymbol{x}_0 = initial\ guess$

$For\ k = 1,2, \ldots$

Evaluate $\mathbf{J} = \boldsymbol{J}(\boldsymbol{x}_k)$

Evaluate $\boldsymbol{f}(\boldsymbol{x}_k)$

Factorization of Jacobian (for example $\mathbf{LU} = \boldsymbol{J}$)

Solve using factorized J (for example $\mathbf{LU}\ \boldsymbol{s}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$)

Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

Newton's method - summary

- Typically quadratic convergence (local convergence)

- Computing the Jacobian matrix requires the equivalent of n^2 function evaluations for a dense problem (where every function of $\boldsymbol{f}(\boldsymbol{x})$ depends on every component of $\boldsymbol{x}$).

- Computation of the Jacobian may be cheaper if the matrix is sparse.

- The cost of calculating the step $\boldsymbol{s}$ is $O(n^3)$ for a dense Jacobian matrix (Factorization + Solve)

- If the same Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_k)$ is reused for several consecutive iterations, the convergence rate will suffer accordingly (trade-off between cost per iteration and number of iterations needed for convergence)

Inverse Kinematics