System Control Engineering

Koichi Hashimoto
Graduate School of Information Sciences

Text: Feedback Systems:
— An Introduction for Scientists and Engineers —
Author: Karl J. Astrom and Richard M. Murray

Text: `http://www.cds.caltech.edu/~murray/amwiki/index.php/Version_2.11b`
Web: `http://www.ic.is.tohoku.ac.jp/en/SystemControlEngineering/`

Today's topics

- Linearization
- Robot Kinematics
- Robot Control
- Examples
- Robot Dynamics

Linearization (Lyapunov's indirect method)

Linearization of Nonlinear Systems

- Consider the nonlinear system

$$\dot{x} = f(x), \qquad f : D \to \mathbb{R}^n$$

and assume that $x = x_e \in D$ is an equilibrium point.

- Taylor series expansion about the equilibrium

$$f(x) = f(x_e) + \left.\frac{\partial f}{\partial x}\right|_{x=x_e} (x - x_e) + \text{h.o.ts}$$

- Neglecting the h.o.ts and recalling $f(x_e) = 0$, we have

$$f(x) = \left.\frac{\partial f}{\partial x}\right|_{x=x_e} (x - x_e)$$

- Now defining

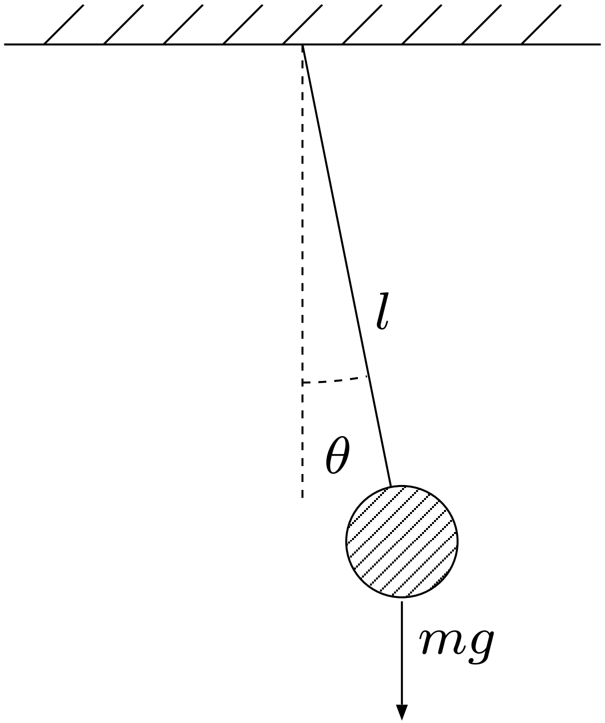
$$\bar{x} = x - x_e, \quad \dot{\bar{x}} = \dot{x}, \quad A = \left.\frac{\partial f}{\partial x}\right|_{x=x_e} = \left.\frac{\partial f}{\partial x}\right|_{\bar{x}=0}$$

we have $\dot{\bar{x}} = A\bar{x}$.

Lyapunov's Indirect Method

- **Theorem 3.11** Let $x = 0$ be an equilibrium point for a nonlinear system $\dot{x} = f(x)$. Assume that A is a matrix obtained by liniarization. Then if the eigenvalues λ_i of the matrix A satisfy Re $\lambda_i < 0$, the origin is an exponentially stable equilibrium point.

Example: Pendulum with friction



- Dynamical equation:

$$ml\ddot{\theta} + mg\sin\theta + bl\dot{\theta} = 0$$

- State variables: $x_1 = \theta, x_2 = \dot{\theta}$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{b}{m}x_2 \end{aligned}$$

- Equilibrium points:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} n\pi \\ 0 \end{bmatrix}, \quad n = 0, \pm 1, \pm 2, \ldots$$

Linearization

- System

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -\frac{g}{l}\sin x_1 - \frac{b}{m}x_2\end{aligned}$$

- Around $[x_1, x_2] = [0, 0]^T$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2}\end{bmatrix}\Bigg|_{x=[0,0]^T} \begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{g}{l} & -\frac{b}{m}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

- Eigenvalues of A:

$$s(s + \frac{b}{m}) + \frac{g}{l} = 0, \quad \text{stable}$$

Linearization

- System

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{b}{m}x_2$$

- Around $[x_1, x_2] = [\pi, 0]^T$

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2}\end{bmatrix}\Bigg|_{x=[\pi,0]^T} \begin{bmatrix}x_1 \\ x_2\end{bmatrix} = \begin{bmatrix}0 & 1 \\ \frac{g}{l} & -\frac{b}{m}\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix}$$

- Eigenvalues of A:

$$s(s + \frac{b}{m}) - \frac{g}{l} = 0, \quad \text{unstable}$$

Feedback Linearization

Feedback Linearization

- **Example 5.1** Consider the first order system

$$\dot{x} = ax^2 + u$$

Is this system stable?

- We look for a state feedback $u = \phi(x)$ that make the equilibrium point at the origin "asymptotically stabe."
- An obvious way is to **cancels** the nonlinear term

$$u = -ax^2 - x$$

to obtain

$$\dot{x} = -x$$

which is **linear** and globally asymptotically stable.

Feedback Linearization

- System

$$\dot{x} = f(x) + g(x)u, \qquad y = h(x)$$

- Derivative

$$\dot{y} = \frac{dh}{dx}\dot{x} = \frac{dh}{dx}f(x) + \frac{dh}{dx}g(x)u = L_f h(x) + L_g h(x)u$$

- Control input

$$u = a(x) + b(x)v$$

- Closed loop

$$\dot{y} = L_f h(x) + L_g h(x)[a(x) + b(x)v]$$

Feedback Linearization

- Closed loop

$$\dot{y} = L_f h(x) + L_g h(x)[a(x) + b(x)v]$$

- Thus, when

$$b(x) = \frac{1}{L_g h(x)}, \quad a(x) = -\frac{L_f h(x)}{L_g h(x)}$$

- We have

$$\dot{y} = v$$

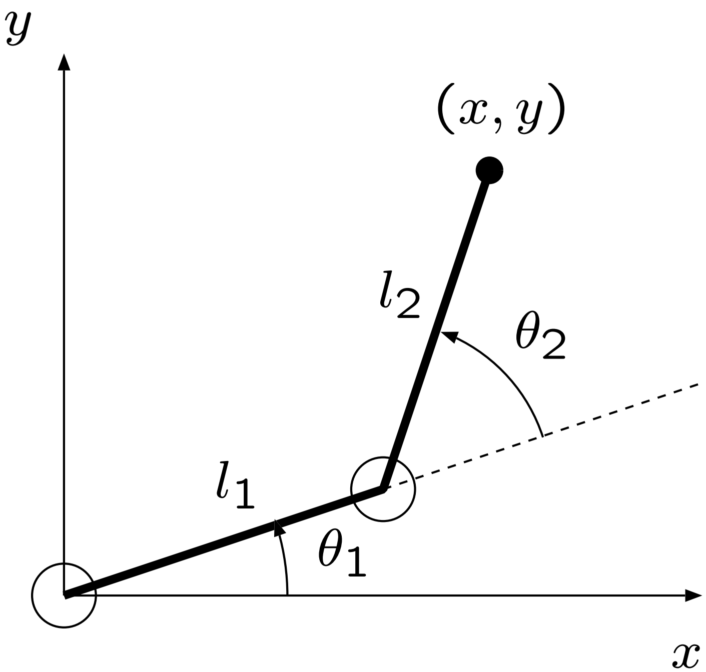
- New input

$$v = y_d - y$$

will stabilize y_d.

Robot Kinematics

Robot System



- Endtip Position:

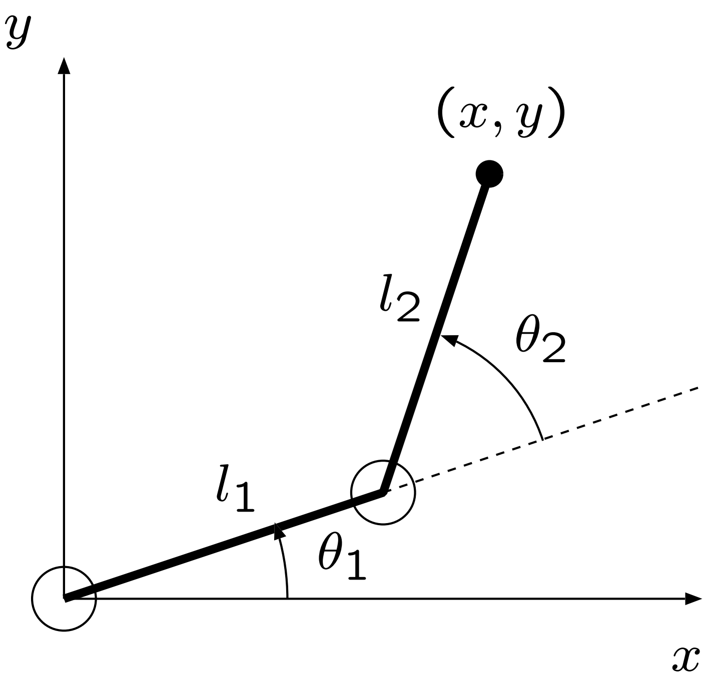
$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{aligned}$$

- Output: $r = (x, y)$
- State: $\theta = (\theta_1, \theta_2)$
- Input:

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}.$$

i.e., the motor driver is velocity control.

Robot Kinematics



- State

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

- Input

$$u = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

- Output

$$r = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Kinematics

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{aligned}$$

Robot State Equation

- State equation (**kinematic nonlinearity**):

$$\begin{aligned}\dot{\theta} &= u \\ r &= g(\theta)\end{aligned}$$

where

$$g(\theta) = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

- Dynamical equation:

$$\begin{aligned}\dot{x} &= -l_1 \sin\theta_1 \dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{y} &= l_1 \cos\theta_1 \dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)\end{aligned}$$

i.e.,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Input-Output Dynamics

- Input-Output (u-r) dynamics

$$\dot{r} = J(\theta)u,$$

where

$$u = \dot{\theta} \quad \text{and} \quad J(\theta) = \frac{\partial g}{\partial \theta} = \begin{bmatrix} \dfrac{\partial x}{\partial \theta_1} & \dfrac{\partial x}{\partial \theta_2} \\ \dfrac{\partial y}{\partial \theta_1} & \dfrac{\partial y}{\partial \theta_2} \end{bmatrix}.$$

- The matrix $J(\theta)$ is called **Jacobi matrix**.

$$J(\theta) = \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

Robot Control

Robot Control

- The objective

$$r = \begin{bmatrix} x \\ y \end{bmatrix} \quad \rightarrow \quad r_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix}.$$

- Let K be a gain matrix and suppose the following velocity control law

$$u = K(r_d - r)$$

where

$$u = \dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad K = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}.$$

- How to choose K?

Feedback linearization

- Closed loop equation:

$$\dot{r} = J(\theta)\dot{\theta} = J(\theta)K(r_d - r)$$

- A typical choice for K is λJ^{-1}, which yields

$$\dot{r} = J(\theta)\dot{\theta} = \lambda J(\theta)J^{-1}(\theta)(r_d - r) = \lambda(r_d - r)$$

- Let $s = r - r_d$ then we have a linearized system

$$\dot{s} = -\lambda s, \qquad s = e^{-\lambda t}s_0$$

and thus

$$s \to 0, \quad \text{i.e.,} \quad r \to r_d \quad (\text{as} \quad t \to \infty)$$

RMRC with Lyapunov

- The control law $K = \lambda J^{-1}(\theta)$, i.e.,

$$\dot{\theta} = \lambda J^{-1}(\theta)(r_d - r)$$

is called **resolved motion rate control**. (Whitney, 1969)

- Consider the following Lyapunov function candidate:

$$V = (r_d - r)^T (r_d - r) \geq 0$$

- Then we have the derivative as follows

$$\begin{aligned} \dot{V} &= -2(r_d - r)^T \dot{r} \\ &= -2(r_d - r)^T J(\theta)\dot{\theta} \\ &= -2(r_d - r)^T J(\theta) K (r_d - r) \\ &= -2(r_d - r)^T (r_d - r) \leq 0 \end{aligned}$$

- $V = 0$ and $\dot{V} = 0$ if and only if $r = r_d$.

RMRC with Lyapunov

- **Theorem 3.2** Let $x = 0$ be an equilibrium point of $\dot{x} = f(x)$, $f : D \to \mathbb{R}^n$, and let $V : D \to \mathbb{R}$ be a continuously differentiable function such that

  (i) $V(0) = 0$,

  (ii) $V(x) > 0 \quad \text{in} \quad D - \{0\}$

  (iii) $\dot{V}(x) < 0 \quad \text{in} \quad D - \{0\}$,

  then $x = 0$ is asymptotically stable.

- When $K = J(\theta)^{-1}$, we have $V(s) > 0$ and $\dot{V}(s) < 0$ for $s = r - r_d \neq 0$ while $V(0) = 0$ and $\dot{V}(0) = 0$.
- Thus RMRC is asymptotically stable in lyapunov's sense.

Robot Control: J_d^{-1}

- Even if the state dependent feedback gain is not possible, we can select

$$K = \lambda J_d^{-1} \qquad \text{where} \qquad J_d^{-1} = \text{inv}(J(\theta_d)) \quad \text{(const.)}$$

where θ_d is the desired joint angle set that satisfies

$$r_d = f(\theta_d)$$

- This choice

$$u = \lambda J_d^{-1}(r_d - r)$$

ensures that

$$\begin{aligned} \dot{V} &= -2(r_d - r)^T J(\theta) K (r_d - r) \\ &= -2\lambda(r_d - r)^T J(\theta) J_d^{-1}(r_d - r) < 0 \end{aligned}$$

around $\theta = \theta_d$ because $J(\theta)J_d^{-1} = I$ at $\theta = \theta_d$.

Robot Control: J_d^T

- The stability is yielded by the positive definiteness of $J(\theta)K$ around $\theta = \theta_d$.
- Another choice

$$u = \lambda J_d^T (r_d - r)$$

can also ensures that

$$\begin{aligned}\dot{V} &= -2(r_d - r)^T J(\theta) K (r_d - r) \\ &= -2\lambda (r_d - r)^T J(\theta) J_d^T (r_d - r) < 0\end{aligned}$$

around $\theta = \theta_d$ because $J(\theta) J_d^T$ is positive definite at $\theta = \theta_d$.

Robot Control: ESM

- Efficient second order minimization

$$u = \lambda \frac{1}{2}(J(\theta) + J_d)^{-1}(r_d - r)$$

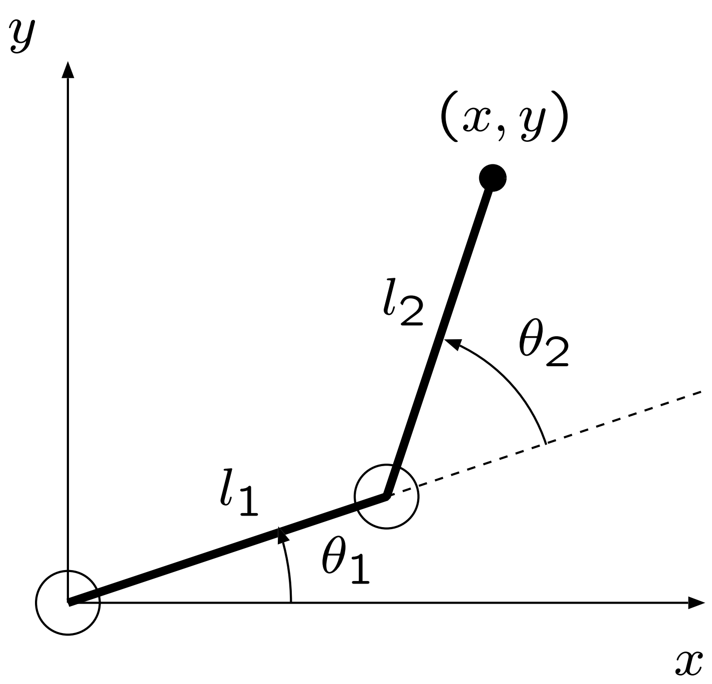
- $J_{\text{esm}} = \frac{1}{2}(J(\theta) + J_d)$ can approximate the Taylor expansion of $r_d - r$ to the second order.

$$\begin{aligned} \dot{V} &= -2(r_d - r)^T J(\theta) K (r_d - r) \\ &= -2\lambda (r_d - r)^T J(\theta) J_{\text{esm}}^{-1} (r_d - r) < 0 \end{aligned}$$

around $\theta = \theta_d$ because $J(\theta) J_{\text{esm}}^{-1} = I$ at $\theta = \theta_d$.

Example

Example



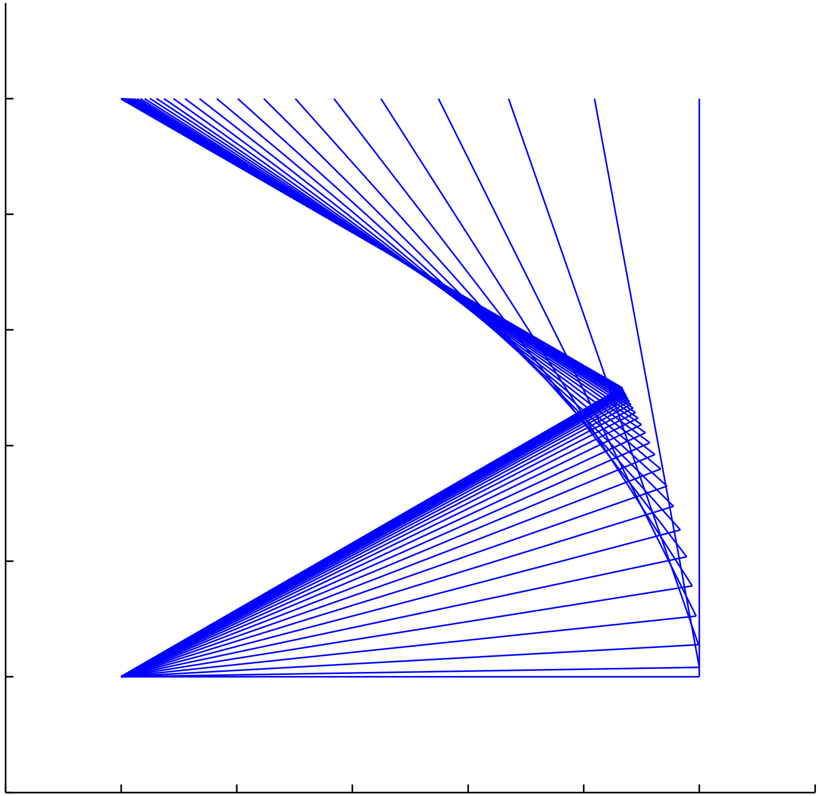
- Suppose that $l_1 = l_2 = 1$
- $r_d = [0, 1]^T$ and $r_0 = [1, 1]^T$
- At the desired position the Jacobi matrix is

$$J_d = \begin{bmatrix} -1 & -1/2 \\ 0 & -\sqrt{3}/2 \end{bmatrix},$$

and

$$J_d^{-1} = \begin{bmatrix} -1 & 1/\sqrt{3} \\ 0 & -2/\sqrt{3} \end{bmatrix}$$

Example: RMRC

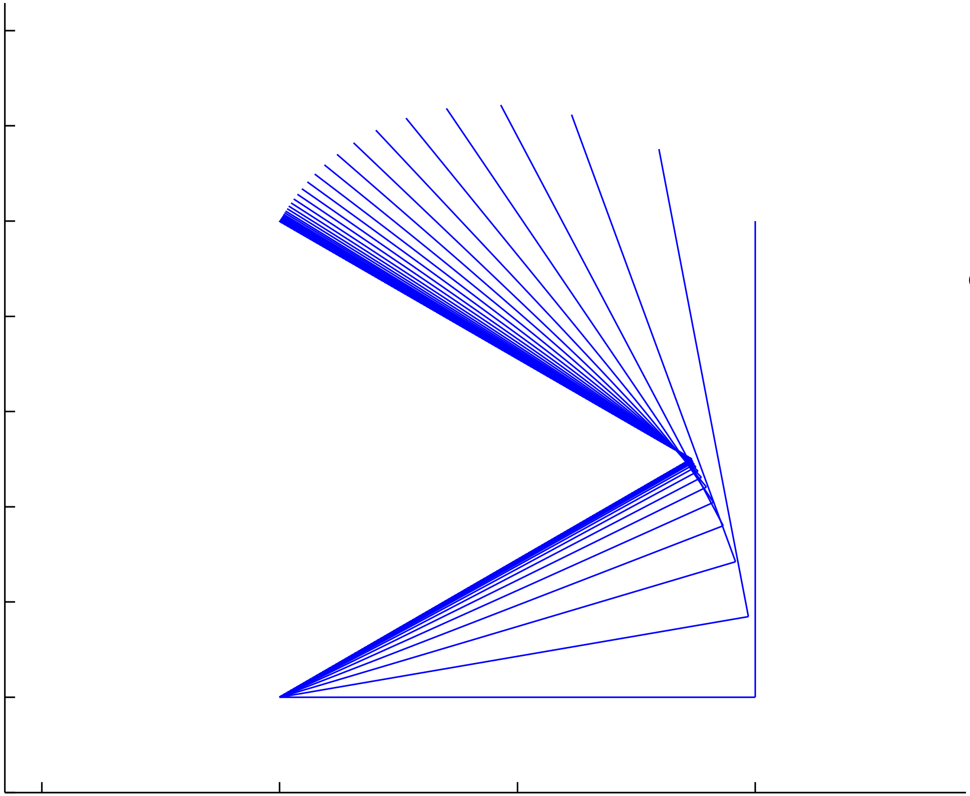


- RMRC yeilds a straight line trajectory

$$\dot{\theta} = \lambda J^{-1}(\theta)(r_d - r)$$
$$\dot{r} = \lambda(r_d - r)$$

$$\sum \Delta\theta_1 = 97.19,$$
$$\sum \Delta\theta_2 = 417.98$$

Example: J_d^{-1}

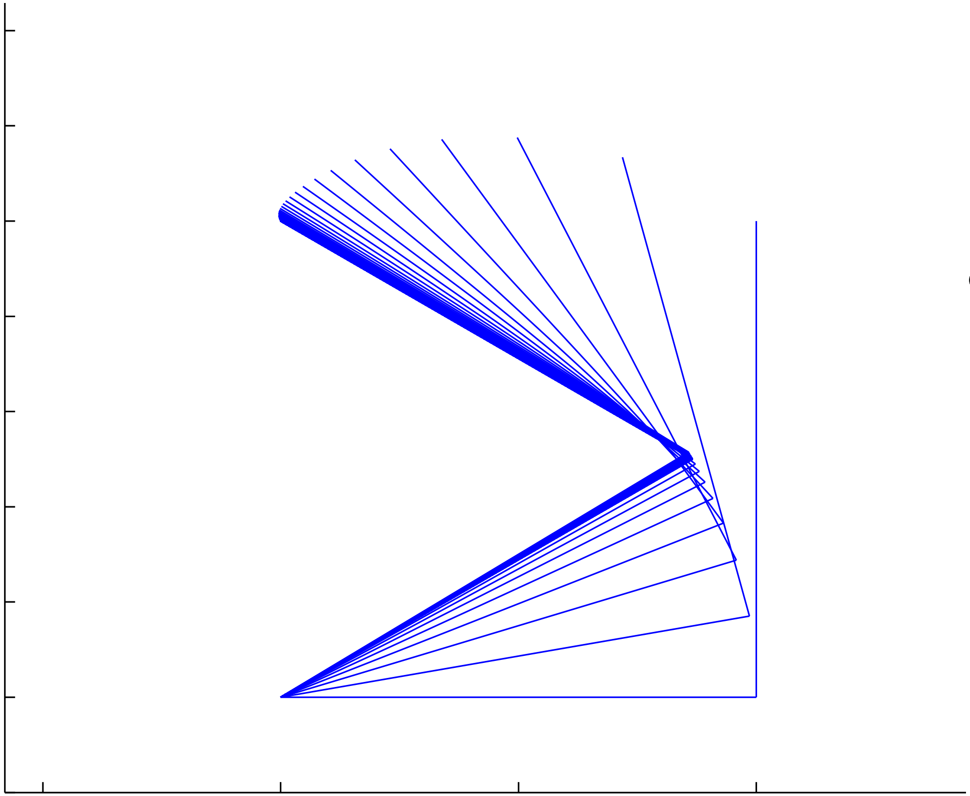


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 = 102.41, \quad \sum \Delta\theta_2 = 413.28$$

Example: J_d^T

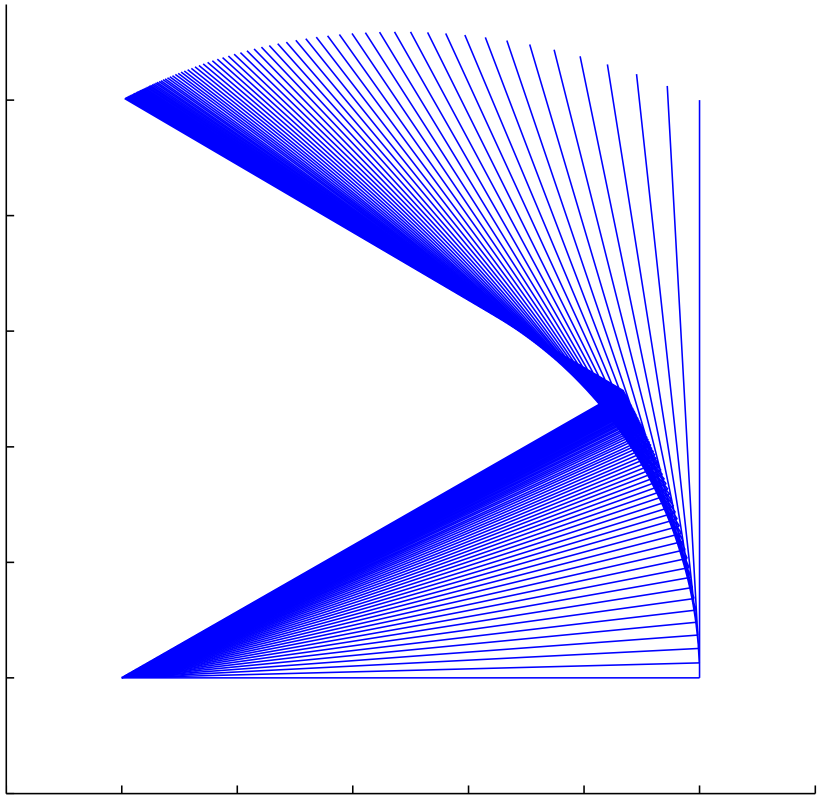


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^T (r_d - r)$$

$$\sum \Delta\theta_1 = 103.22,$$
$$\sum \Delta\theta_2 = 415.68$$

Example: ESM

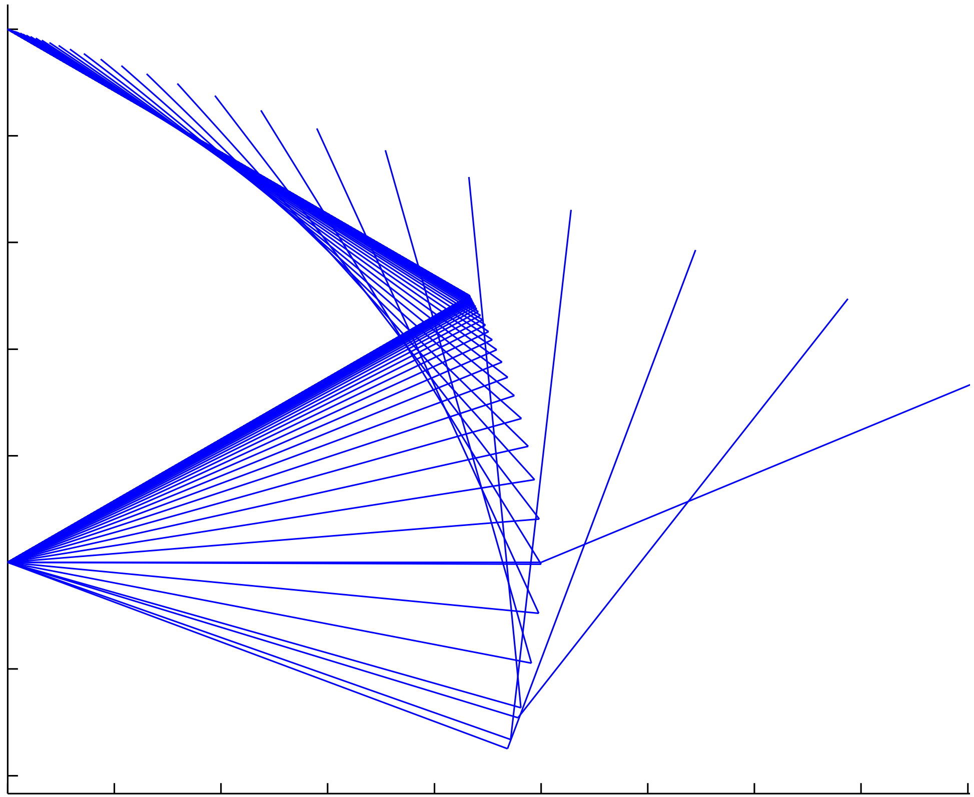


- $J_{\text{esm}} = (J(\theta) + J(\theta_d))/2$

$$\dot{\theta} = \lambda J_{\text{esm}}^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 = 84.33,$$
$$\sum \Delta\theta_2 = 401.02$$

Example: RMRC

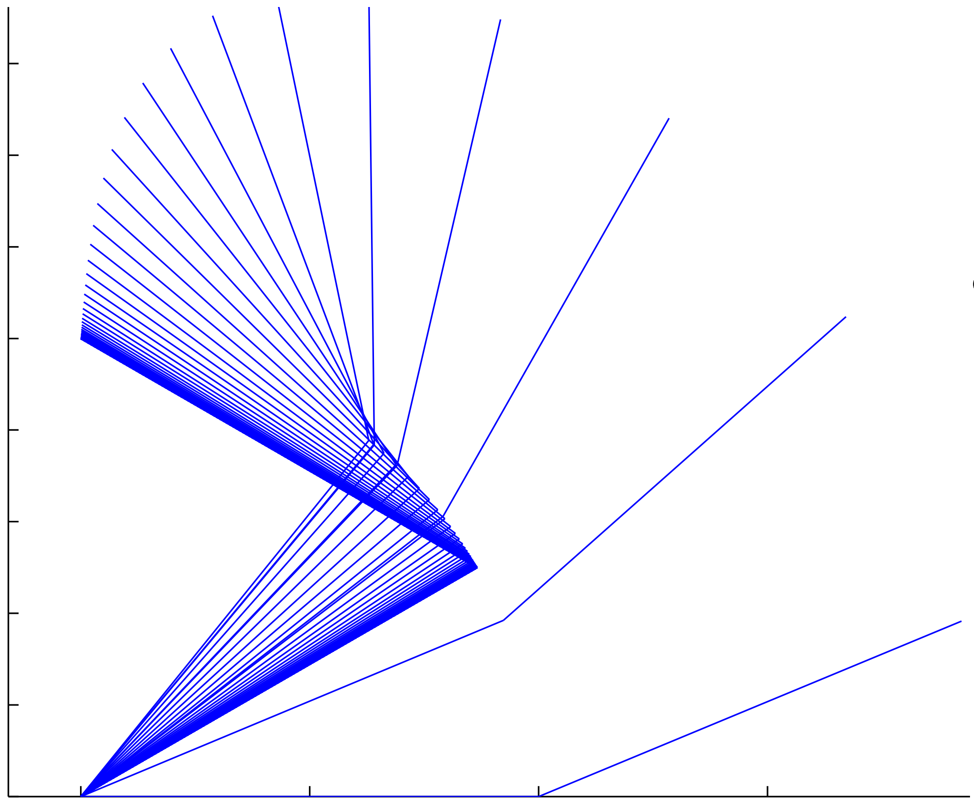


- RMRC yeilds a straight line trajectory

$$\dot{\theta} = \lambda J^{-1}(\theta)(r_d - r)$$
$$\dot{r} = \lambda(r_d - r)$$

$$\sum \Delta\theta_1 = 95.33,$$
$$\sum \Delta\theta_2 = 415.82$$

Example: J_d^{-1}

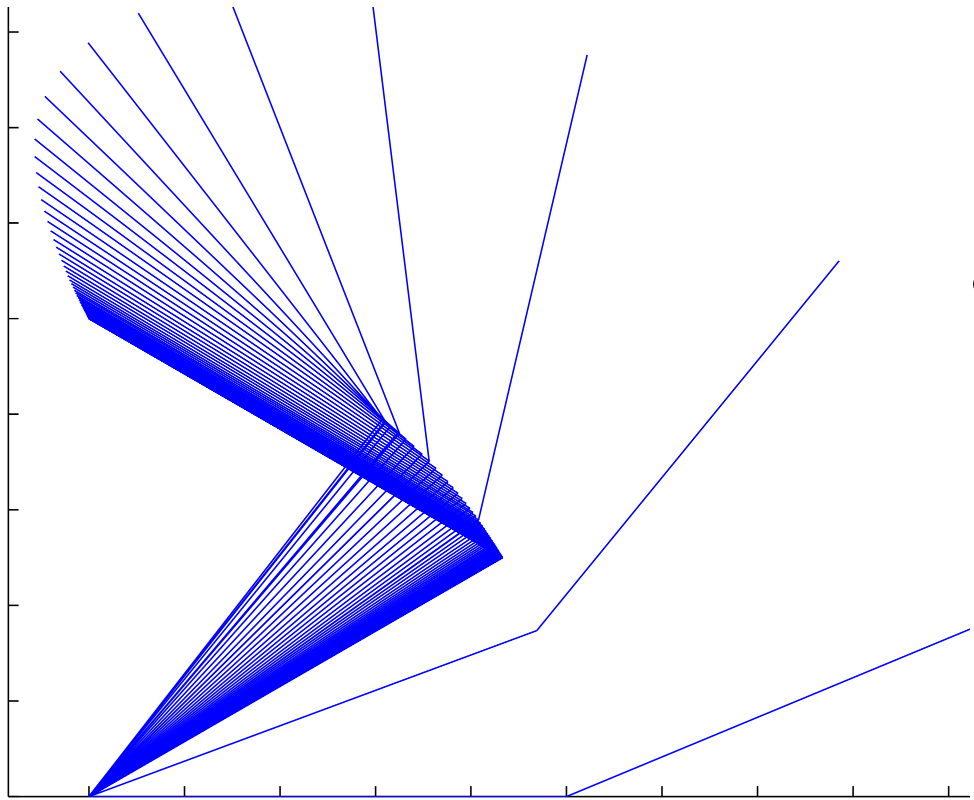


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 = 111.53, \quad \sum \Delta\theta_2 = 390.55$$

Example: J_d^T

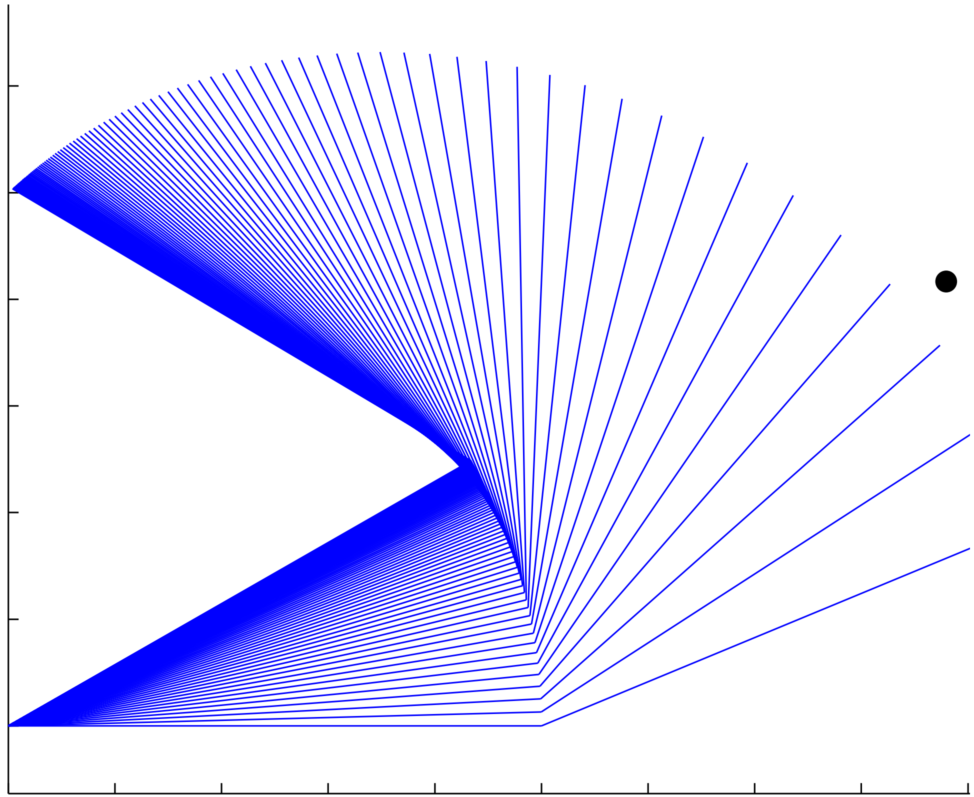


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^T (r_d - r)$$

$$\sum \Delta\theta_1 \;=\; 116.79,$$
$$\sum \Delta\theta_2 \;=\; 391.83$$

Example: ESM

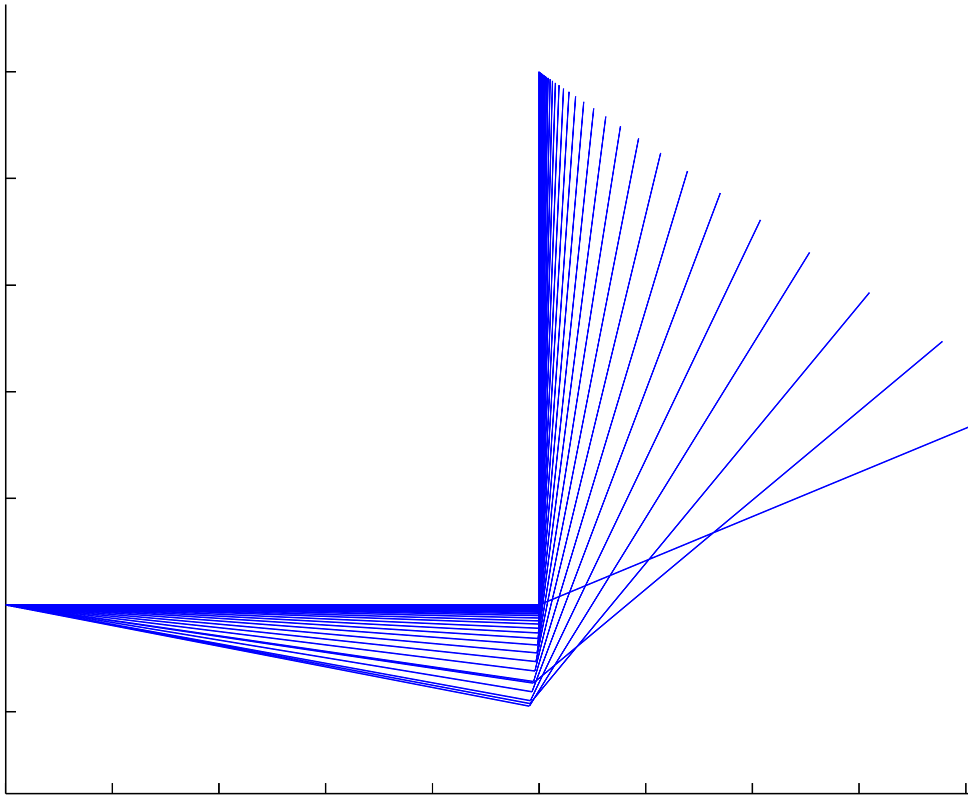


- $J_{\text{esm}} = (J(\theta) + J(\theta_d))/2$

$$\dot{\theta} = \lambda J_{\text{esm}}^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 = 84.33,$$
$$\sum \Delta\theta_2 = 365.12$$

Example: RMRC

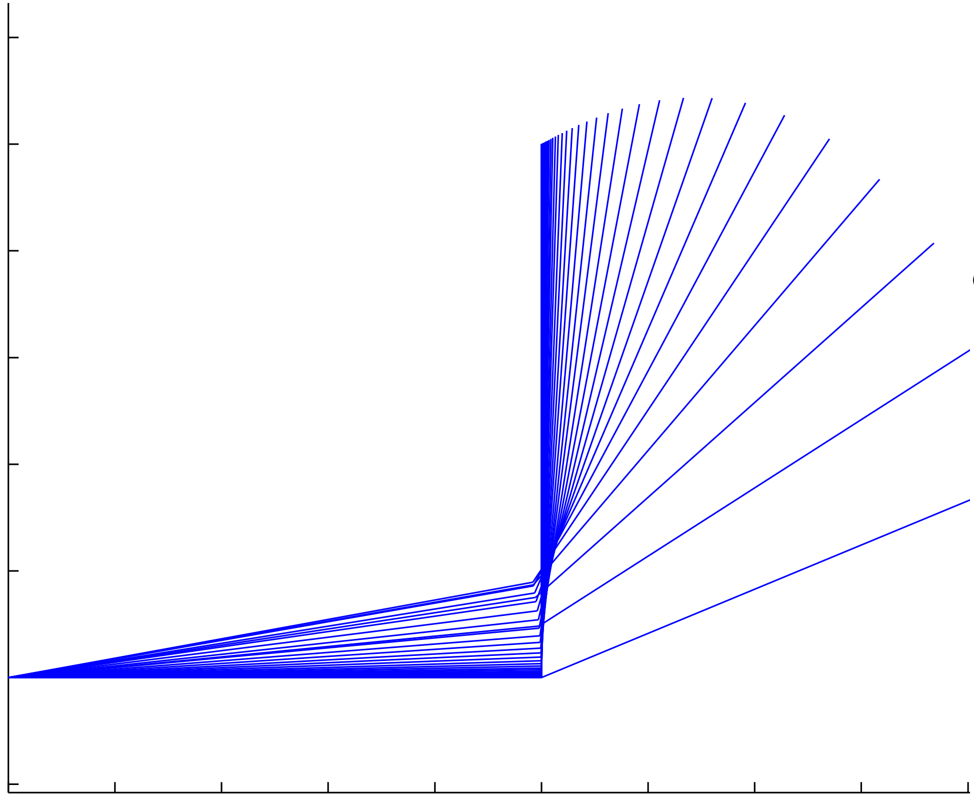


- RMRC yeilds a straight line trajectory

$$\dot{\theta} = \lambda J^{-1}(\theta)(r_d - r)$$
$$\dot{r} = \lambda(r_d - r)$$

$$\sum \Delta\theta_1 = 3.57,$$
$$\sum \Delta\theta_2 = 308.58$$

Example: J_d^{-1}

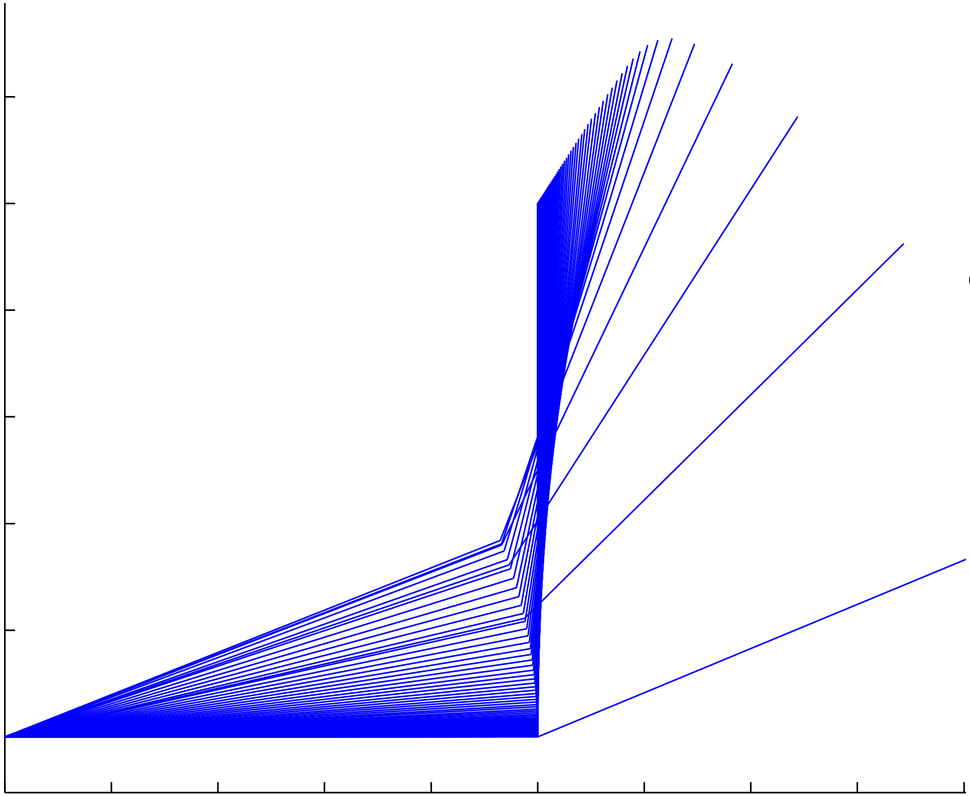


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 = 3.75,$$
$$\sum \Delta\theta_2 = 298.35$$

Example: J_d^T

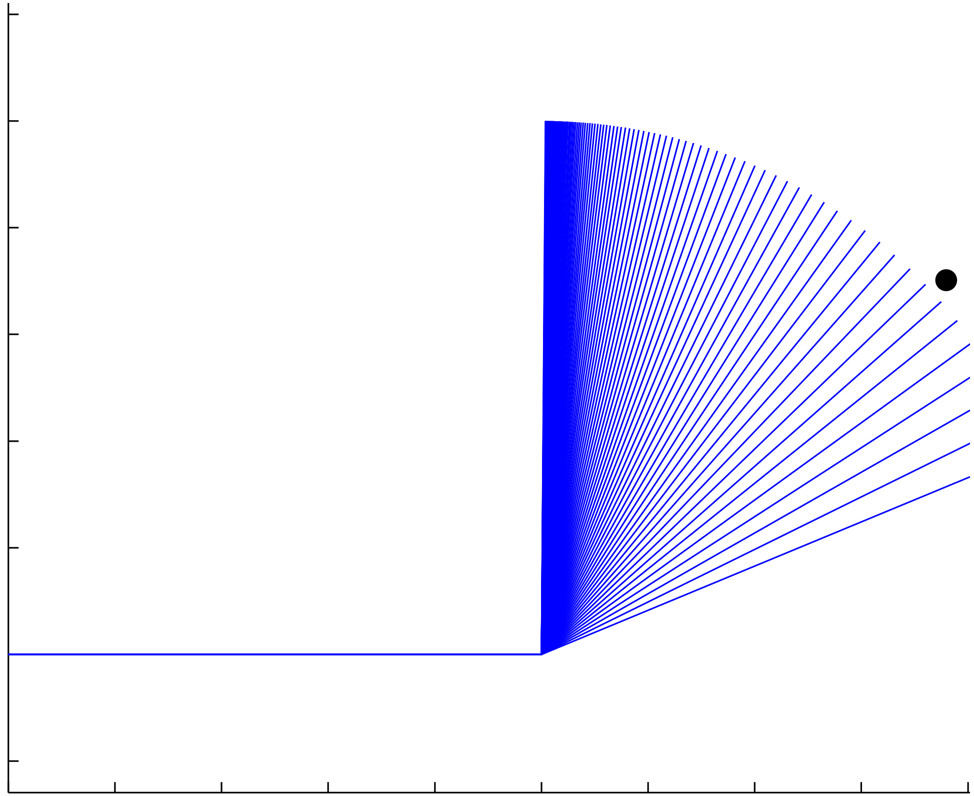


- $J_d = J(\theta_d)$ (const.)

$$\dot{\theta} = \lambda J_d^T (r_d - r)$$

$$\sum \Delta\theta_1 = 14.39,$$
$$\sum \Delta\theta_2 = 286.80$$

Example: ESM



- $J_{\mathsf{esm}} = (J(\theta) + J(\theta_d))/2$

$$\dot{\theta} = \lambda J_{\mathsf{esm}}^{-1}(r_d - r)$$

$$\sum \Delta\theta_1 \;=\; 0.00,$$
$$\sum \Delta\theta_2 \;=\; 270.15$$

MATLAB Simulation Code

```
method='4';
global l1;  global l2;
l1=1;       l2=1;
q0=[0;pi/8];  r0=kin(q0);         Jac0=Jac(q0);
rd=[1;1];     qd=invkin(rd,q0);   Jacd=Jac(qd);

t=0:0.1:20;   dq=zeros(length(t),2);
hold off; clf;
switch method
    case '1'
        K=inv(Jacd);
        dq=twolinksim(q0,rd,K,t,'JT');
    case '2'
        K=Jacd';
```

```
        dq=twolinksim(q0,rd,K,t,'JT');
    case '3'
        K=[];
        dq=twolinksim(q0,rd,K,t,'Ji');
    case '4'
        K=Jacd;
        dq=twolinksim(q0,rd,K,t,'esm');
    otherwise
        sprintf('no mthod defined: %s', method);
end
sum(abs(dq))
```

MATLAB Simulation Code


```
function dq=twolinksim(q0,rd,K,t,method)
    global l1;
    global l2;

    switch method
    case 'Ji'
        [~,dq]=ode45(@rob0,t,q0,[],rd,K);
    case 'esm'
        [~,dq]=ode45(@rob1,t,q0,[],rd,K);
    otherwise
        [~,dq]=ode45(@rob,t,q0,[],rd,K);
    end
    x1=l1*cos(dq(:,1));
    x2=x1+l2*cos(dq(:,1)+dq(:,2));
```

```
    y1=l1*sin(dq(:,1));
    y2=y1+l2*sin(dq(:,1)+dq(:,2));
    hold on
    N=length(dq(:,1));
    for i = 1:2:N
        plot([0,x1(i)],[0,y1(i)]);
        plot([x1(i),x2(i)],[y1(i),y2(i)]);
    end
end
```

MATLAB Simulation Code


```
function dq=rob(t,q,rd,K)
  r=kin(q);
  dq=K*(rd-r);
end

function dq=rob0(t,q,rd,K)
  r=kin(q);
  Ja=Jac(q);
  dq=Ja\(rd-r);
end

function dq=rob1(t,q,rd,K)
  r=kin(q);
  Ja=Jac(q);
  dq=(K+Ja)\(rd-r)/2;
end
```

Robot Dynamics

- It is well know that robot system in general has the dynamical equation of the form

$$M(\theta)\ddot{\theta} + C(\dot{\theta}, \theta) + D\dot{\theta} + P(\theta) = \tau$$

where $M(\theta)$ is inertia, $C(\dot{\theta}, \theta)$ is centrifugal and Coriolis force, D is friction coefficient, and $P(\theta)$ is potential.

- When we have the estimates of these parameters, then a control input

$$\tau = \widehat{M}(\theta)v + \widehat{C}(\dot{\theta}, \theta) + \widehat{D}\dot{\theta} + \widehat{P}(\theta)$$

where

$$v = \ddot{\theta}_d + k_2(\dot{\theta}_d - \dot{\theta}) + k_1(\theta_d - \theta)$$

will linearize and stabilize the trajectory $\theta = \theta_d$.

Robot Dynamics

- If the parameters are exactly known then substituting τ in the dynamical equation yields

$$\ddot{e} + k_2\dot{e} + k_1 e = 0$$

where $e = \theta_d - \theta$.

- This control scheme is called **inverse dynamics** or **resolved motion acceleration control**. (J Luh, M Walker, R Paul, 1980)