Tutorial: Active Inference Controller for Dynamic Systems

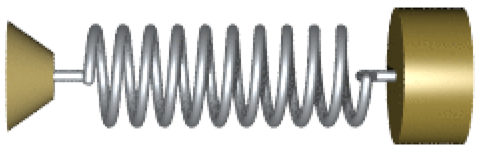
Ajith Anil Meera

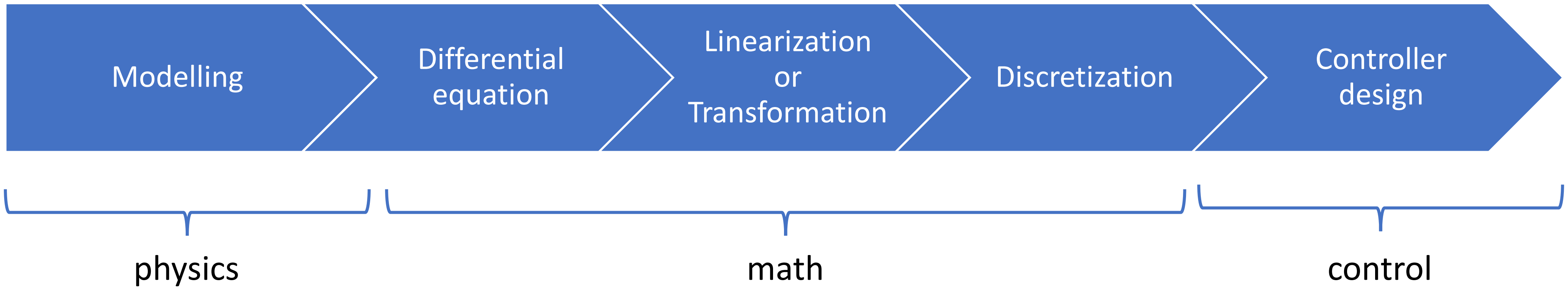
Post doctoral researcher,
Radboud University, The Netherlands

Active inference for robots

Lanillos, P., Meo, C., Pezzato, C., Meera, A. A., Baioumy, M., Ohata, W., Tschantz, A., Millidge, B., Wisse, M., Buckley, C. L. & Tani, J. (2021). **Active Inference in Robotics and Artificial Agents: Survey and Challenges**. *arXiv preprint arXiv:2112.01871*.

Methodology

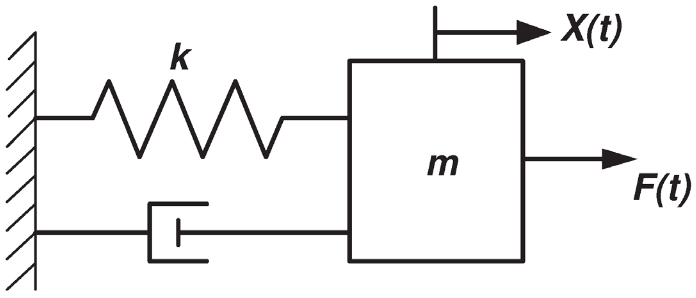


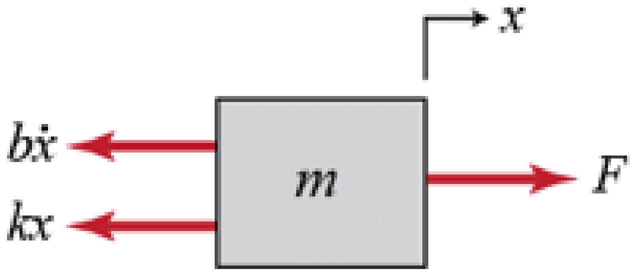



```
Discretization: sys_d = c2d(ss(model.A,model.B,model.C,[]),dt,'zoh’);
Controller:     dFda = (brain.x - goal_x).'*Pi_g*(-pinv(model.A)*model.B) + Pa*a(:,i-1);
                dFdaa = (-pinv(model.A)*model.B).'*Pi_g*(-pinv(model.A)*model.B) + Pa;
                 a(:,i) = a(:,i-1) + (expm(-k*dFdaa*dt)-eye(ny))*pinv(dFdaa)*dFda;
Generative process: model.x(:,i+1) = sys_d.A*model.x(:,i) + sys_d.B*a(:,i);
```

Modelling the dynamic system

Step 1: Derive the equations of motion of the dynamic system





Free body diagram

$$F - b\dot{x} - kx = 0$$

static system

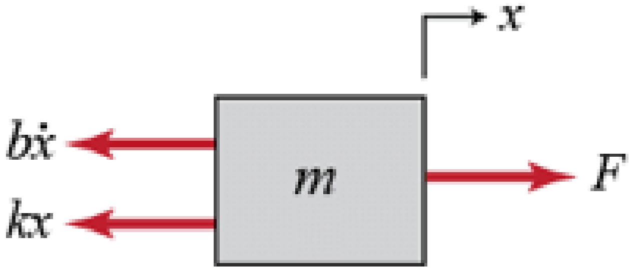
$$F - b\dot{x} - kx = m\ddot{x}$$

dynamic system

Dynamic system → Governing equations of motion

Equations of motion to differential equations

Step 2: convert the dynamic model to a differential equation



$$F - b\dot{x} - kx = m\ddot{x}$$

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = F$$

$$\frac{d^2x}{dt^2} + \frac{b}{m}\frac{dx}{dt} + \frac{k}{m}x = \frac{F}{m}$$

Governing equation of motion $\longrightarrow$ Differential equations of motion

Linear time invariant (LTI) state space systems

Step 3a: convert the differential equation to an LTI system

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

A, B, C, D are independent of time

$$\frac{d^2x}{dt^2} + \frac{b}{m}\frac{dx}{dt} + \frac{k}{m}x = \frac{F}{m}$$

$$\dot{X} = AX + Bu$$

$$x_1 = x \qquad x_2 = \dot{x}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{x} \end{bmatrix}$$

$$\ddot{x} = -\frac{b}{m}\dot{x} - \frac{k}{m}x + \frac{F}{m}$$

$$\dot{X} = \begin{bmatrix} x_2 \\ -\frac{b}{m}\dot{x} - \frac{k}{m}x + \frac{F}{m} \end{bmatrix}$$

Linear time invariant state space systems

$$x_1 = x \qquad x_2 = \dot{x}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{x} \end{bmatrix}$$

$$\ddot{x} = -\frac{b}{m}\dot{x} - \frac{k}{m}x + \frac{F}{m}$$

$$\dot{X} = \begin{bmatrix} x_2 \\ -\frac{b}{m}\dot{x} - \frac{k}{m}x + \frac{F}{m} \end{bmatrix}$$

$$\dot{X} = \begin{bmatrix} x_2 \\ -\frac{b}{m}\dot{x} - \frac{k}{m}x + \frac{F}{m} \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{b}{m}x_2 - \frac{k}{m}x_1 + \frac{F}{m} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} F$$

$$\dot{X} = AX + Bu$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \qquad u = F, \qquad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Linear time invariant state space systems

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

$$\dot{X} = AX + Bu$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \quad u = F, \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$y = CX \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

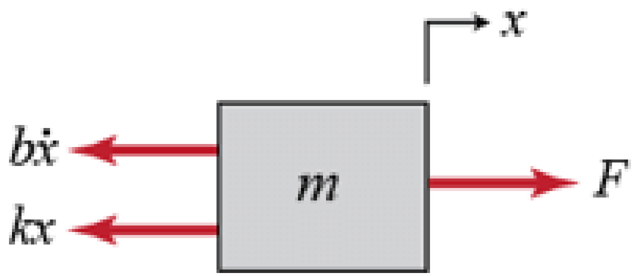
Observation model

Differential equations $\longrightarrow$ LTI system

LTI systems through linearization

Step 3b: linearize the differential equation to form an LTI system

$$\dot{X} = \begin{bmatrix} x_2 \\ -\frac{b}{m}x_2 - \frac{k}{m}x_1 + \frac{F}{m} \end{bmatrix} = \begin{bmatrix} f_1(X,u) \\ f_2(X,u) \end{bmatrix} \qquad y = x_1 = g(X,u)$$



$$\begin{aligned} \dot{X} &= f(X,u) \\ y &= g(X,u) \end{aligned} \qquad \begin{aligned} \ddot{X} &= f(X,u) = AX + Bu \\ y &= g(X,u) = CX + Du \end{aligned}$$

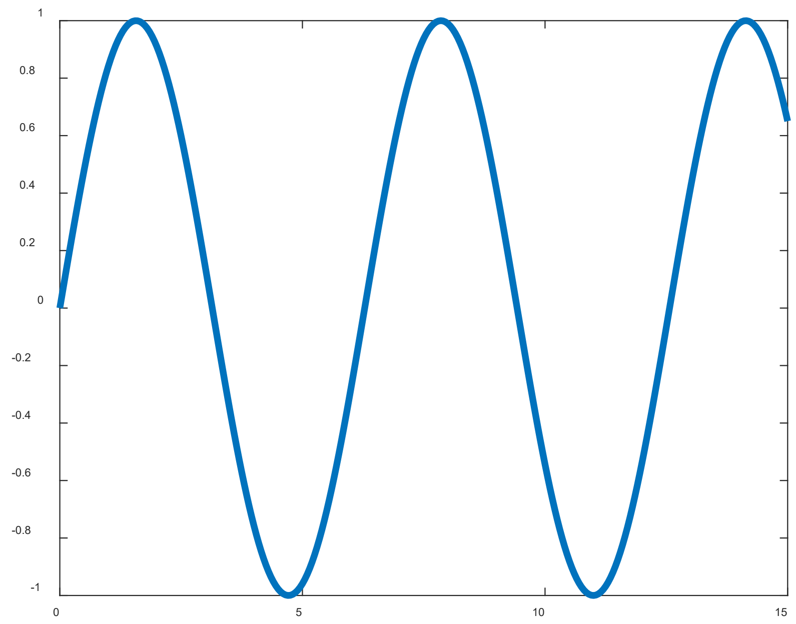
$$A = \frac{\partial f}{\partial X}_{\{X=X^e,u=u^e\}}, \qquad B = \frac{\partial f}{\partial u}_{\{X=X^e,u=u^e\}}, \qquad C = \frac{\partial g}{\partial X}_{\{X=X^e,u=u^e\}}, \qquad D = \frac{\partial g}{\partial u}_{\{X=X^e,u=u^e\}}$$

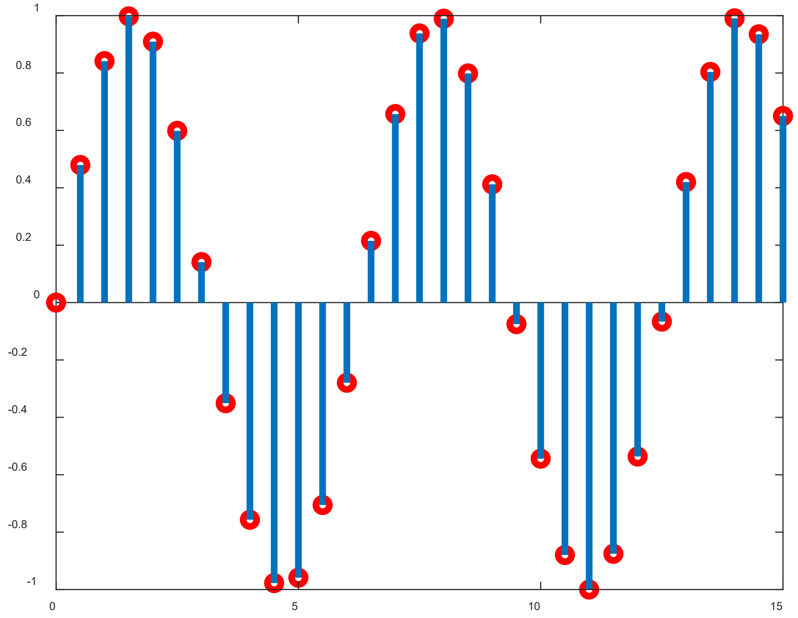
$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \qquad B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \qquad C = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} \end{bmatrix} = [1 \quad 0], \qquad D = \frac{\partial g}{\partial u} = 0$$

Discretization

Step 4: discretize the continuous time model to discrete time model

Why? Because the robot world is discrete!!





Discretization

$$\dot{X} = A_c X + B_c u$$

$$x[t+1] = A_d x[t] + B_d u[t]$$

$$y[t] = C_d x[t] + D_d u[t]$$

$$A_d = e^{A_c \Delta t}$$

$$B_d = {A_c}^{-1}(A_d - I)B_c$$

$$C_d = C_c, \qquad D_d = D_c$$

$e^{A_c \Delta t}$ is a matrix exponential expm(.) and not exp(.)

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$$

Stability of an LTI system

Step 5: check the stability of the system

The system is stable if the output is finite for all possible finite inputs

$$\dot{X} = A_c X + B_c u$$

The system is asymptotically stable if and only if all the eigen values of A_c are in the left-hand plane

$$\dot{X} = A_d X + B_d u$$

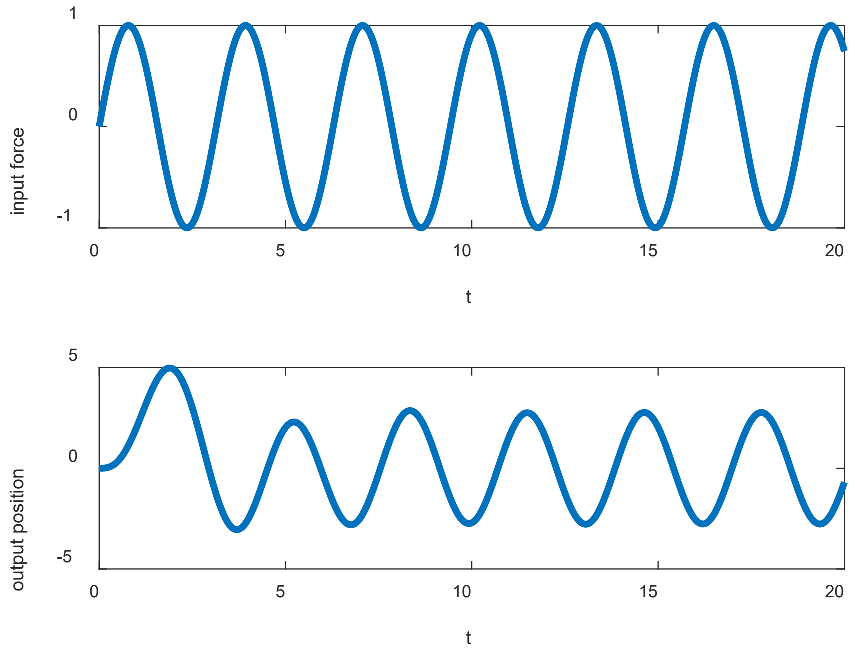
The system is asymptotically stable if and only if all the eigen values of A_d are inside the unit circle

Stability of an SMD system

$$A_c = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}$$

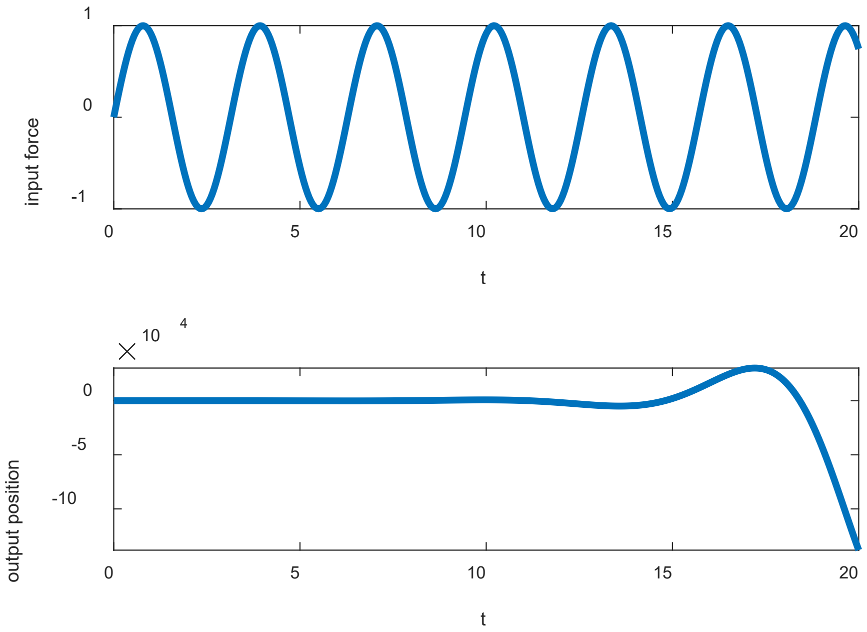
$$m = 0.1kg, \qquad k = 0.1\frac{N}{m}, \qquad b = 0.1\frac{Ns}{m},$$

$$eig(A_c) = \{-0.5 + 0.866i, -0.5 - 0.866i\}$$



$$m = 0.1kg, \qquad k = 0.1\frac{N}{m}, \qquad b = -0.1\frac{Ns}{m},$$

$$eig(A_c) = \{0.5 + 0.866i, 0.5 - 0.866i\}$$



Active Inference controller

Step 6: derive the control law

- Take control actions that follow the gradient of free energy

$$\frac{du}{dt} = -\gamma \frac{\partial F}{\partial u}$$

F – free energy
γ – learning rate
u – control action

$$F = \frac{1}{2}(\tau - \tau^g)^T P^{\tau^g}(\tau - \tau^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

τ – task variable
τ^g - goal for task variable
P^{τ^g} - goal precision (inverse covariance matrix) of τ^g
η^u - prior on control action
P^u - prior control precision

Active Inference controller

$$\frac{du}{dt} = -\gamma \frac{\partial F}{\partial u}$$

$$F = \frac{1}{2}(\tau - \tau^g)^T P^{\tau g}(\tau - \tau^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

$\eta^u = 0$ with high P^u forces the controller to minimize the control action
$\eta^u = 0$ with low P^u allows the controller to explore

$$\frac{du}{dt} = -\gamma \frac{\partial F}{\partial \tau}\frac{\partial \tau}{\partial u} + \frac{\partial F}{\partial u}$$

$$\frac{\partial F}{\partial \tau} = (\tau - \tau^g)^T P^{\tau g}$$

$$\frac{\partial F}{\partial u} = (u - \eta^u)^T P^u$$

System dynamics

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Active inference control for position control

$$\frac{du}{dt} = -\gamma \frac{\partial F}{\partial \tau}\frac{\partial \tau}{\partial u} \qquad \tau = x \qquad \frac{du}{dt} = -\gamma \frac{\partial F}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial F}{\partial u}$$

$$F = \frac{1}{2}(x - x^g)^T P^{x^g}(x - x^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

$$\dot{x} = Ax + Bu$$

$$\frac{\partial \dot{x}}{\partial u} = A\frac{\partial x}{\partial u} + B, \qquad \frac{\partial \dot{x}}{\partial u} = 0 \; \Rightarrow \frac{\partial x}{\partial u} = -A^{-1}B$$

$$\frac{du}{dt} = -\gamma\left(\frac{\partial F}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial F}{\partial u}\right) = -\gamma\left((x - x^g)^T P^{x^g} A^{-1}B + u^T P^u\right)$$

Ajith Anil Meera and Pablo Lanillos, "*Towards metacognitive robot decision making for tool selection*" IWAI 2023.

Active inference control for velocity control

$$\frac{du}{dt} = -\gamma \frac{\partial F}{\partial \tau}\frac{\partial \tau}{\partial u} \qquad \tau = \dot{x} \qquad \frac{du}{dt} = -\gamma \frac{\partial F}{\partial x}\frac{\partial \dot{x}}{\partial u} + \frac{\partial F}{\partial u}$$

$$F = \frac{1}{2}(\dot{x} - \dot{x}^g)^T P^{\dot{x}^g}(\dot{x} - \dot{x}^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

$$\dot{x} = Ax + Bu$$

$$\frac{\partial \dot{x}}{\partial u} = A\frac{\partial x}{\partial u} + B, \qquad \frac{\partial x}{\partial u} = \begin{bmatrix}1\\1\end{bmatrix} \Rightarrow \frac{\partial \dot{x}}{\partial u} = A\begin{bmatrix}1\\1\end{bmatrix} + B$$

$$\frac{du}{dt} = -\gamma\left(\frac{\partial F}{\partial x}\frac{\partial \dot{x}}{\partial u} + \frac{\partial F}{\partial u}\right) = -\gamma\left((\dot{x} - \dot{x}^g)^T P^{\dot{x}^g}\left(A\begin{bmatrix}1\\1\end{bmatrix} + B\right) + u^T P^u\right)$$

Ajith Anil Meera and Pablo Lanillos, “*Towards metacognitive robot decision making for tool selection*” IWAI 2023.

Discretization of the controller

$$u(t+\Delta t) = u(t) - \gamma \frac{\partial F}{\partial u} \Delta t$$

Euler method

Step 7: discretize the control law for discrete time update rule

$$u(t+\Delta t) = u(t) + \left(e^{\left(-k\frac{\partial^2 F}{\partial u^2}\Delta t\right)} - I\right)\left(\frac{\partial^2 F}{\partial u^2}\right)^{-1} \frac{\partial F}{\partial u}$$

Unknowns: $\frac{\partial F}{\partial u}, \frac{\partial^2 F}{\partial u^2}$

<u>Position control:</u>

$$F = \frac{1}{2}(x - x^g)^T P^{x^g}(x - x^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

$$\frac{\partial F}{\partial u} = (x - x^g)^T P^{x^g} A^{-1} B + u^T P^u$$

$$\frac{\partial^2 F}{\partial u^2} = (A^{-1}B)^T P^{x^g} A^{-1} B + P^u$$

<u>Velocity control:</u>

$$F = \frac{1}{2}(\dot{x} - \dot{x}^g)^T P^{\dot{x}^g}(\dot{x} - \dot{x}^g) \quad + \frac{1}{2}(u - \eta^u)^T P^u (u - \eta^u)$$

$$\frac{\partial F}{\partial u} = (\dot{x} - \dot{x}^g)^T P^{\dot{x}^g} \left(A \begin{bmatrix}1\\1\end{bmatrix} + B\right) + u^T P^u$$

$$\frac{\partial^2 F}{\partial u^2} = \left(A \begin{bmatrix}1\\1\end{bmatrix} + B\right)^T P^{\dot{x}^g} \left(A \begin{bmatrix}1\\1\end{bmatrix} + B\right) + P^u$$

Pitfalls of discretization

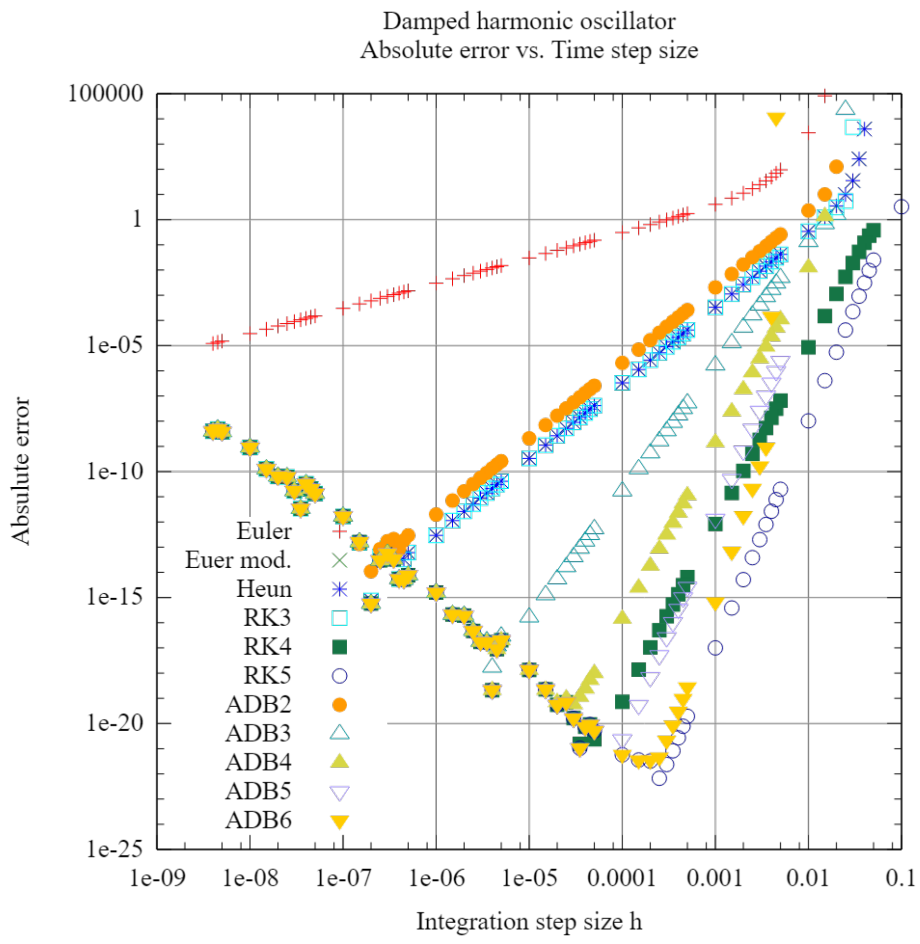


Image source: https://beltoforion.de/en/runge-kutta_vs_euler/

Active inference controller for dynamic systems

- Step 1: model the motion of the system
- Step 2: convert the dynamic model to a differential equation
- Step 3a: convert the differential equation to an LTI system
- Step 3b: linearize the differential equation to form an LTI system
- Step 4: discretize the continuous time model to discrete time model
- Step 5: check the stability of the system
- Step 6: derive the control law
- Step 7: discretize the control law for discrete time update rule

MATLAB implementation

```
% Define the model
model.A = [0 1; -k/m -b/m];                model.B = [0; 1/m];        model.C = [1 0];

% define all the required variables
Pa = .00001*eye(nu);     Pi_g = diag([.01 .01]);    goal_x = [.5; 0];    k_h = 1; dt = .01; nt = 2000;

% discretize the system
sys_d = c2d(ss(model.A,model.B,model.C,[]),dt,'zoh');

for i = 1:nt
        dFda = (brain.x - goal_x).'*Pi_g*(-pinv(model.A)*model.B) + Pa*a(:,i-1);
        dFdaa = (-pinv(model.A)*model.B).'*Pi_g*(-pinv(model.A)*model.B) + Pa;
        a(:,i) = a(:,i-1) + (expm(-k*dFdaa*dt)-eye(ny))*pinv(dFdaa)*dFda;

        % Generative process - take action in the world
        model.x(:,i+1) = sys_d.A*model.x(:,i) + sys_d.B*a(:,i);
end
```

Running code: https://github.com/ajitham123/mTAIC_IWAI2023

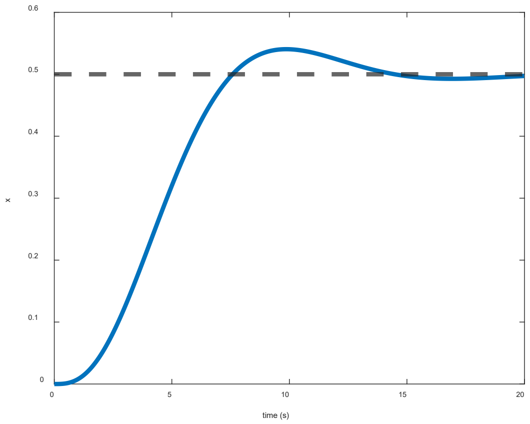
Ajith Anil Meera and Pablo Lanillos, "*Towards metacognitive robot decision making for tool selection*" IWAI 2023.

Active inference to control an SMD

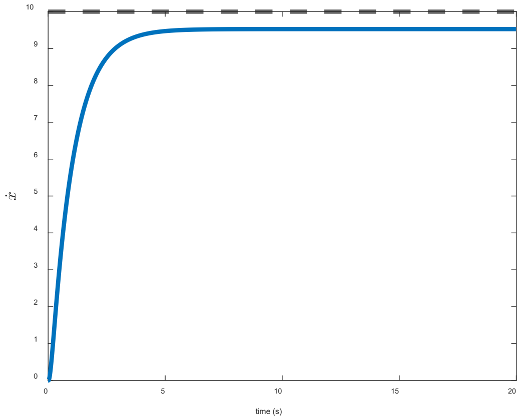
Position control

velocity control

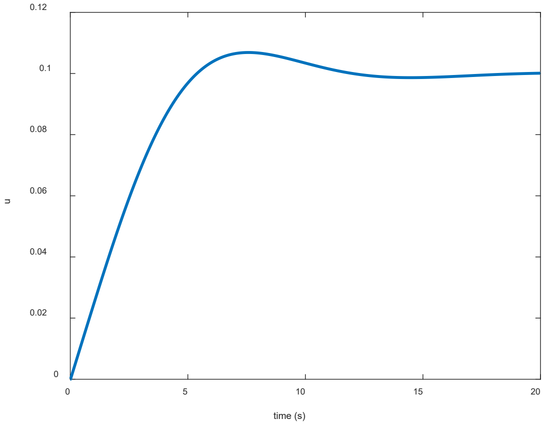
Position



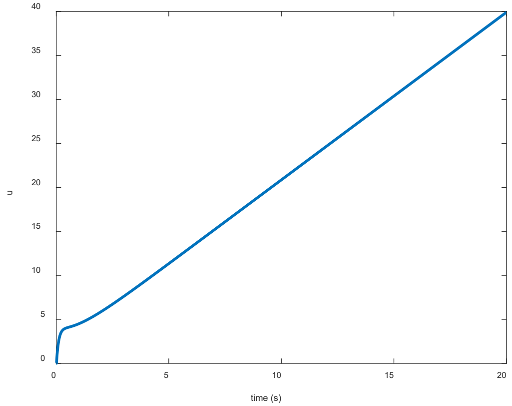
velocity



Control
force



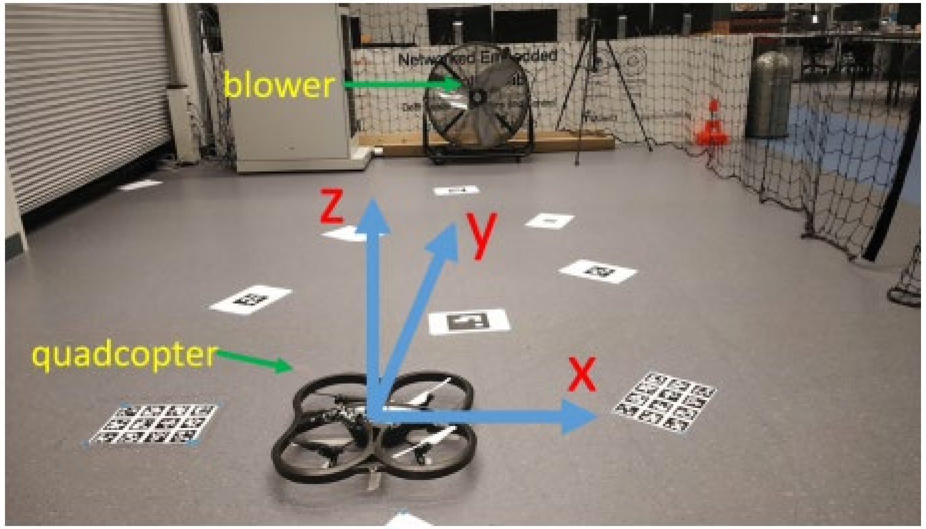
Control
force



Ajith Anil Meera and Pablo Lanillos, "*Towards metacognitive robot decision making for tool selection*" IWAI 2023.

Active inference for robots

Modelling: Quadrotors as LTI system



$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{v} + \mathbf{w}$$
$$\mathbf{y} = C\mathbf{x} + \mathbf{z}.$$

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{c_{B\phi}}{I_{xx}} & -\frac{c_{B\phi}}{I_{xx}} & -\frac{c_{B\phi}}{I_{xx}} & \frac{c_{B\phi}}{I_{xx}} \end{bmatrix} \begin{bmatrix} pwm_1 \\ pwm_2 \\ pwm_3 \\ pwm_4 \end{bmatrix}$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix}$$

ϕ and $\dot{\phi}$ are roll angle and roll velocity
Pwm are the input signals to the rotors

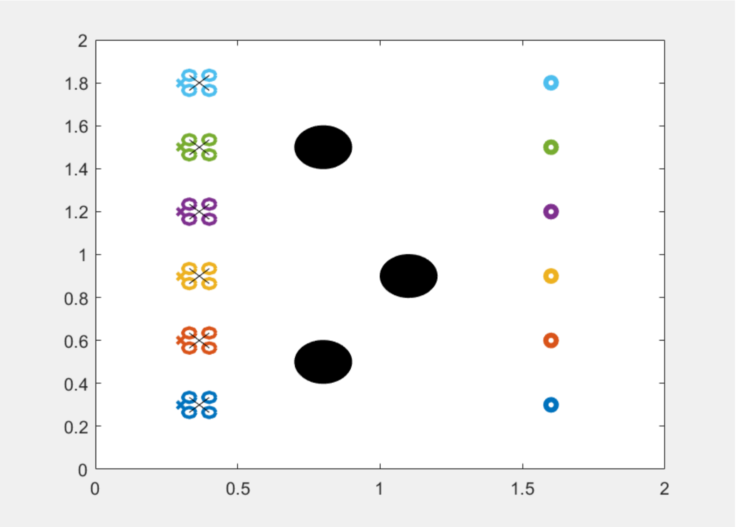
Bos, Fred, et al. "Free energy principle for state and input estimation of a quadcopter flying in wind." *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

Active inference for robot navigation

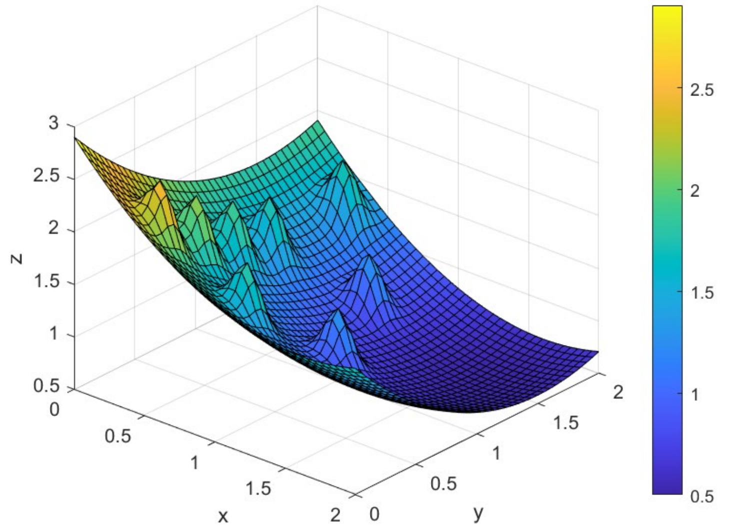
$$F^g = \frac{1}{2}(x - x^g)^T P^{x^g}(x - x^g) \quad + \frac{1}{2}(y - y^g)^T P^{y^g}(y - y^g)$$

$$\frac{\partial u^x}{\partial t} = -\gamma \frac{\partial F}{\partial x}\frac{\partial x}{\partial u^x} = -\gamma \frac{\partial F}{\partial x}$$

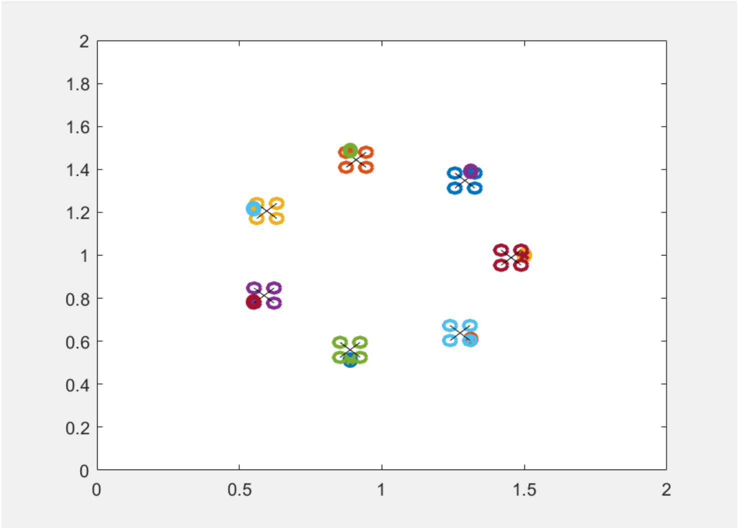
$$F = F^g + F^{os} + F^{od}$$



Static obstacles



Free energy

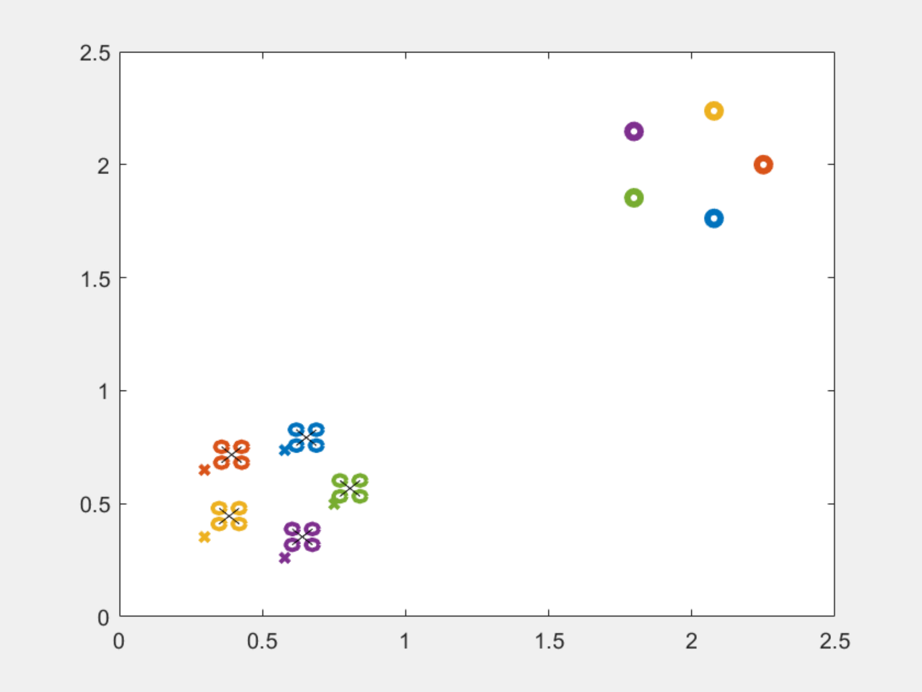


Dynamic obstacles

"Free Energy Principle Based Precision Modulation for Robot Attention: towards brain inspired robot intelligence", PhD thesis, Department of Cognitive Robotics, TU Delft, 2023

Multi robot navigation in formation

- Expects to reach the goal
- Expects to avoid obstacles
- Expects to keep the formation



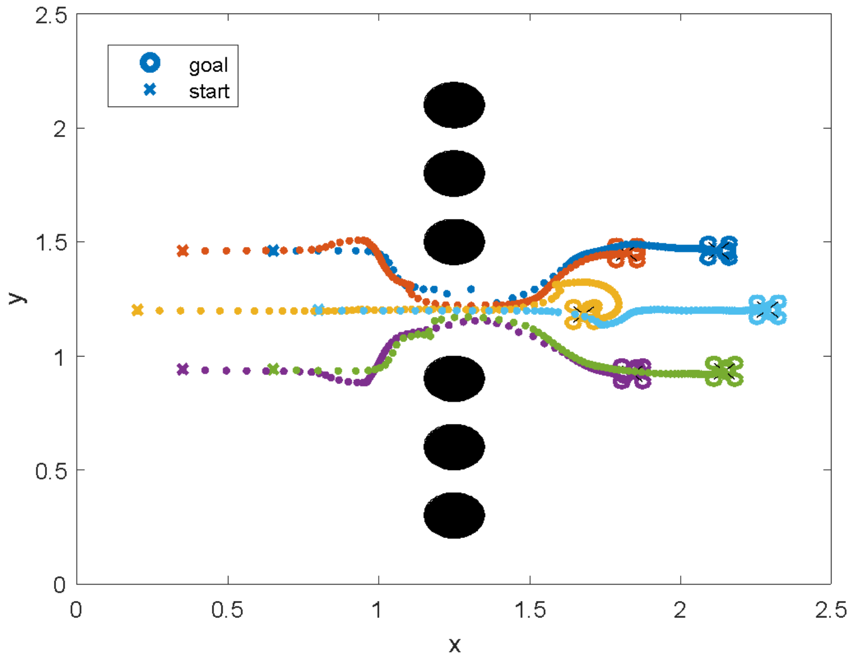
$$F = F^g + F^{os} + F^{od} + F^f$$

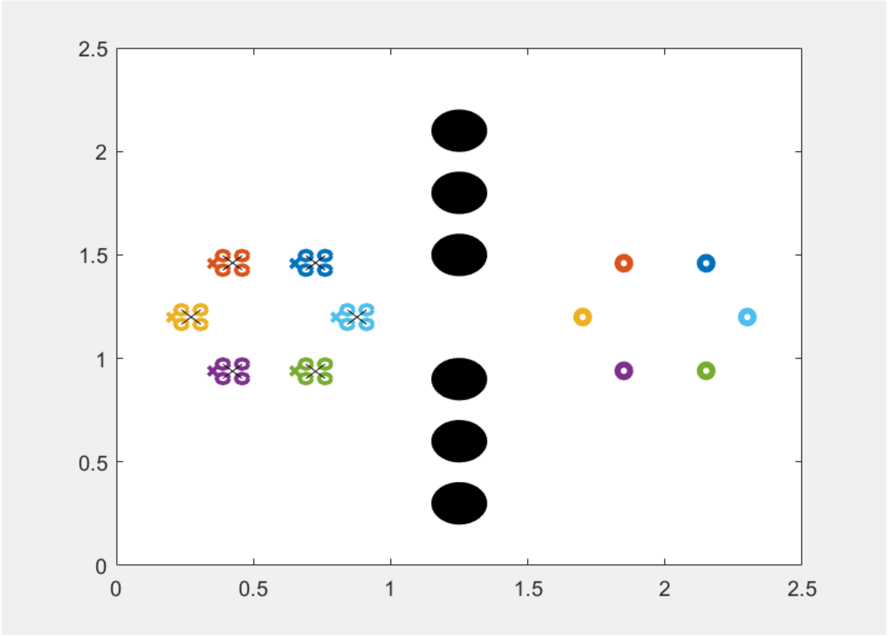
$$F^f = \frac{1}{2}\Pi^f \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\|p^j - p^i\| - \|pf^j - pf^i\| \right)^2$$

p^i $and p^j$ are position vector of drone i and j
p^{fi} $and p^{fj}$ are the initial position vector of drone i and j

"Free Energy Principle Based Precision Modulation for Robot Attention: towards brain inspired robot intelligence", PhD thesis, Department of Cognitive Robotics, TU Delft, 2023

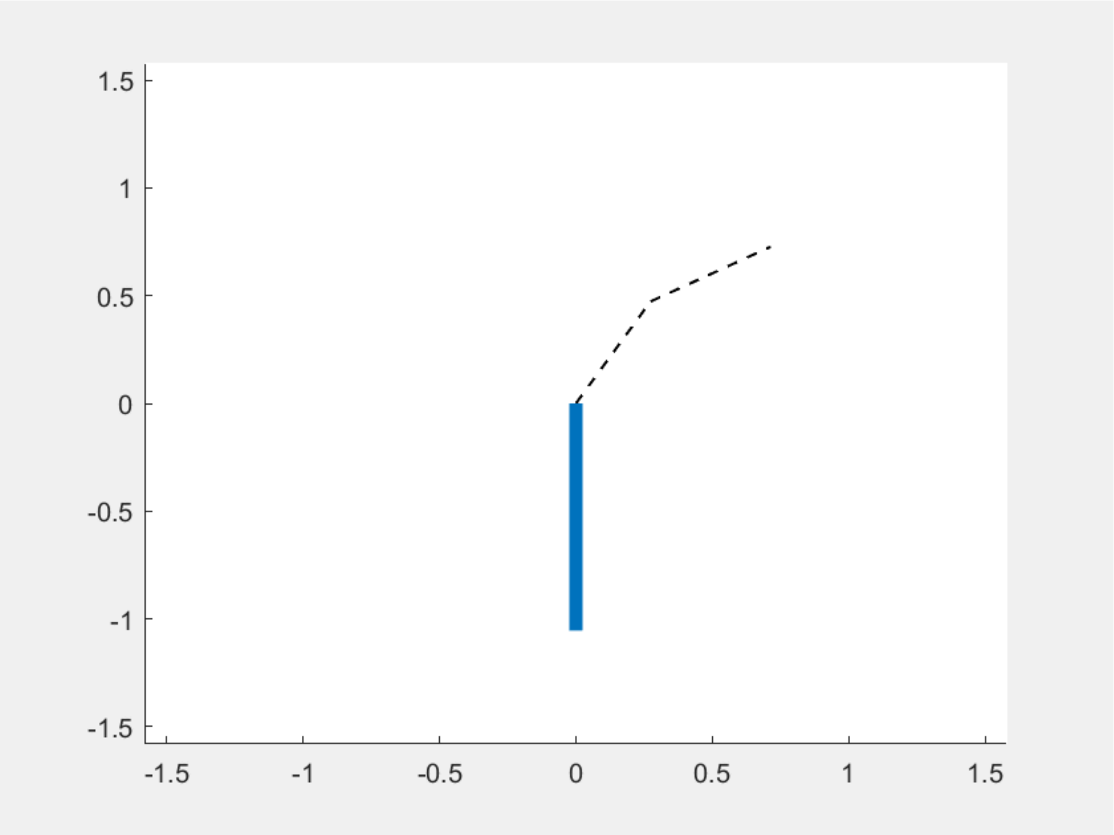
Escape manoeuvre for tight spaces





"Free Energy Principle Based Precision Modulation for Robot Attention: towards brain inspired robot intelligence", PhD thesis, Department of Cognitive Robotics, TU Delft, 2023

Active Inference on a 2DOF robot arm



Ajith Anil Meera and Pablo Lanillos, "*Towards metacognitive robot decision making for tool selection*" IWAI 2023.

Thank you

Martijn Wisse, TU Delft

Pablo Lanillos, Radboud University