Trajectory Learning via Stable Dynamical Systems

Group 4

Dynamic Movement Primitives

- make robots faithfully mimic human motions
- DMP - one of the most used frameworks for trajectory learning from a single demonstration
- based on a system of second-order ODEs
- forcing term can be “learned” to encode the desired trajectory

A DMP is a system of second-order ODEs of mass–spring–damper type with a forcing term:

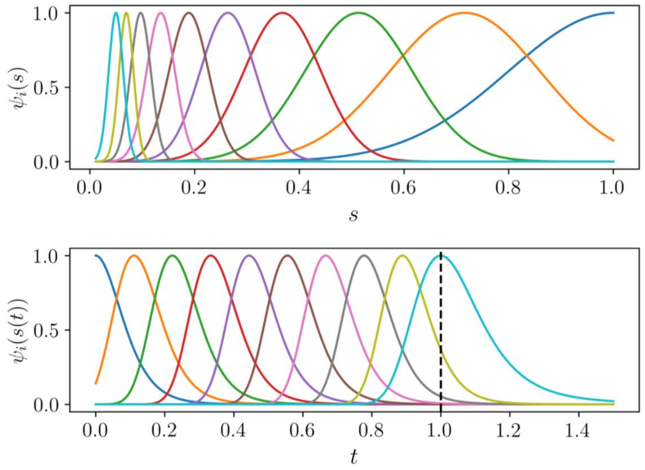
$$\tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s)$$
$$\tau\dot{\mathbf{x}} = \mathbf{v}$$
$$\tau\dot{s} = -\alpha s$$

forcing term

DMP system

$$\tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s)$$
$$\tau\dot{\mathbf{x}} = \mathbf{v}$$
$$\tau\dot{s} = -\alpha s$$

$$f(s) = \frac{\sum_{i=0}^{N} \omega_i \psi_i(s)}{\sum_{i=0}^{N} \psi_i(s)} s$$



Learning:

- set a trajectory
- find weights

Execution:

- solve numerically the differential problem

Learning weights

$$f(s) = \sum_{i=0}^{N} \omega_i \underbrace{\left(\frac{\psi_i(s)}{\sum_{j=0}^{N} \psi_j(s)} s \right)}_{\Psi_i(s)}$$

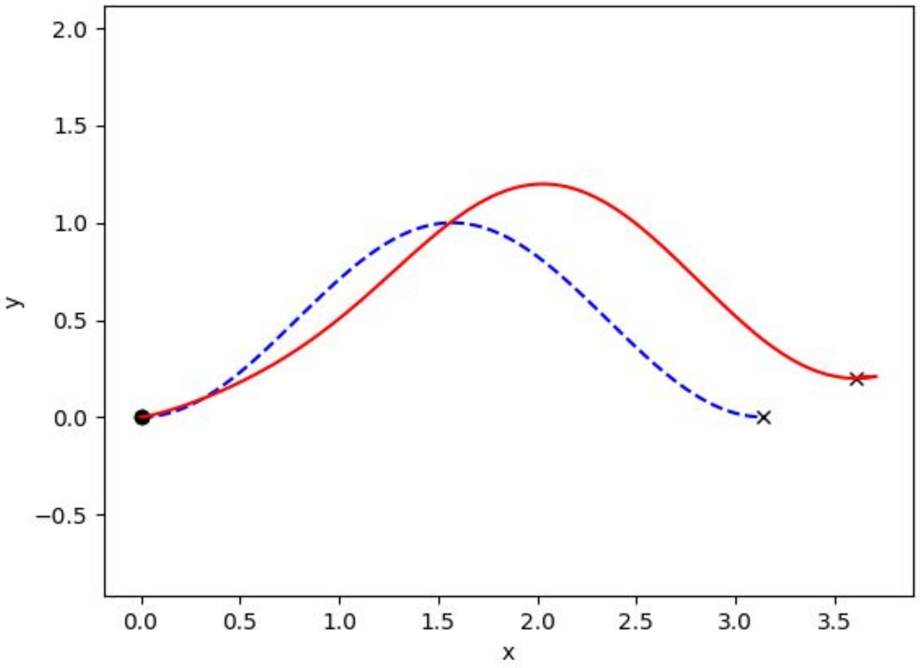
After having estimated the velocity and acceleration of the desired trajectory (e.g. with the Finite Differences method), we can learn the weights by re-writing the forcing term as

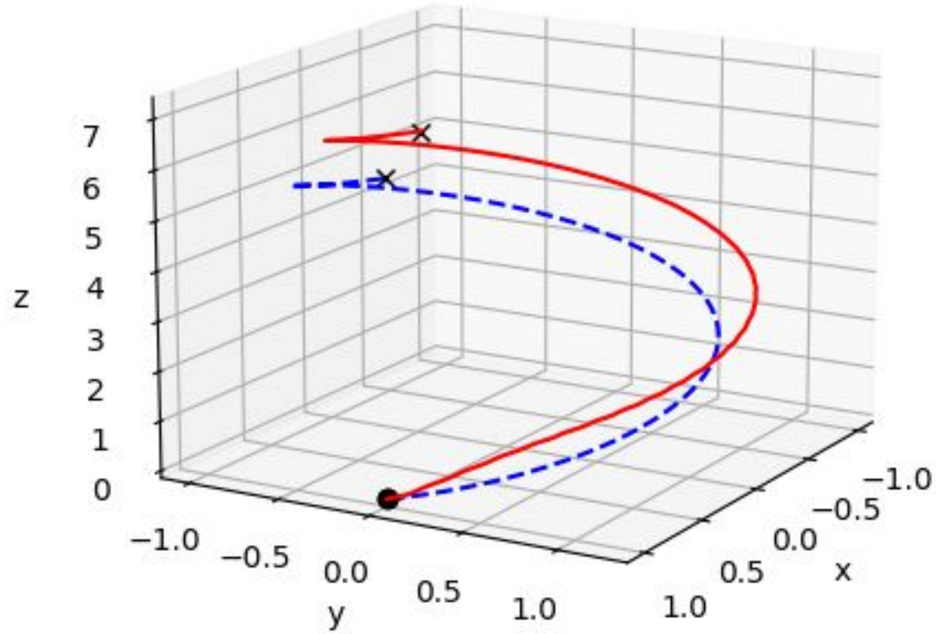
$$\tilde{f}(s) = K^{-1}(\dot{\tilde{v}} + D\tilde{v}) - (g - \tilde{x}) + (g - \tilde{x}_0)s$$

and solving the regression problem for the weights matrix

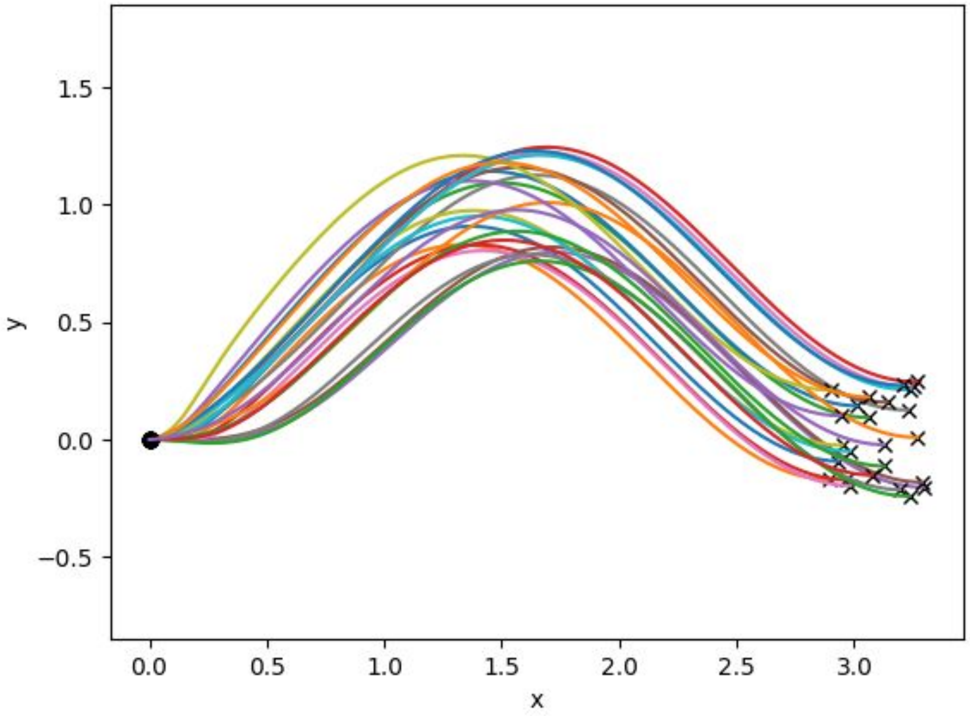
$$\left[\tilde{f}(s_0) \ \cdots \ \tilde{f}(s_N) \right] = \Omega \left[\Psi(s_0) \ \cdots \ \Psi(s_N) \right]$$

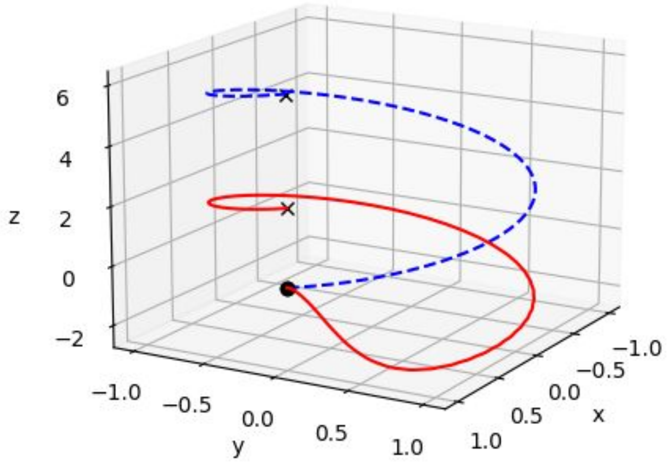
Examples of DMPs

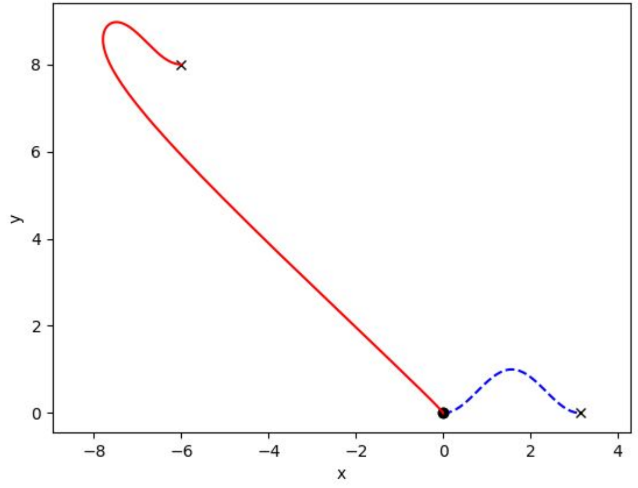




Examples of DMPs







Scalability

Theorem (Hoffmann et al.). *DMPs are invariant under affine transformations of the coordinate system.*

This property allows to obtain an invariance under different reference frames as long as we are able to define an invertible matrix **S** such that

$$\mathbf{g}' - \mathbf{x}'_0 = \mathbf{S}(\mathbf{g} - \mathbf{x}_0).$$

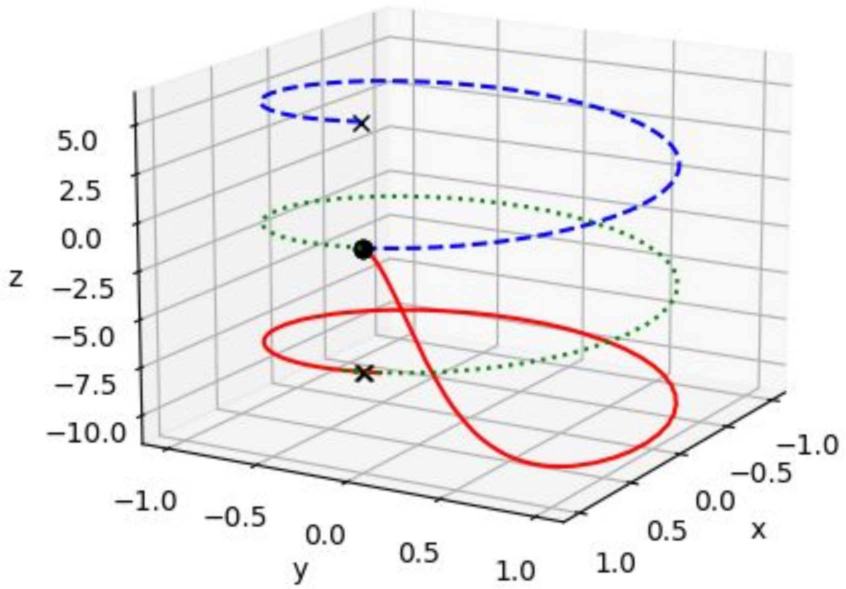
The new forces will then be

$$\mathbf{f} = \mathbf{S}\mathbf{f}'$$

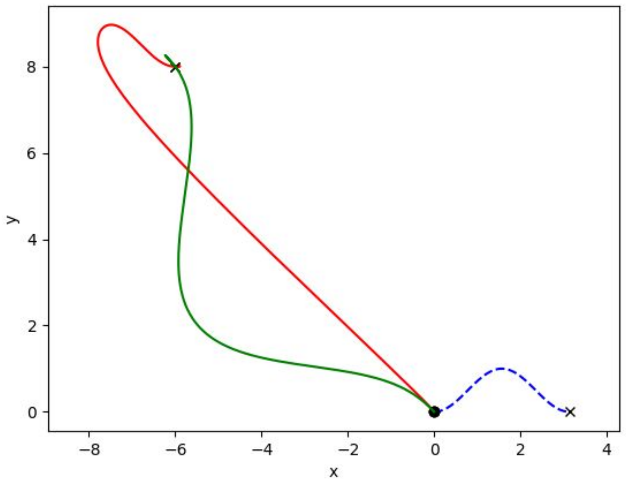
For instance, **S** can be the roto-dilational matrix

$$\mathbf{S} = \frac{\|\mathbf{g}' - \mathbf{x}'_0\|}{\|\mathbf{g} - \mathbf{x}_0\|}\mathbf{R}$$

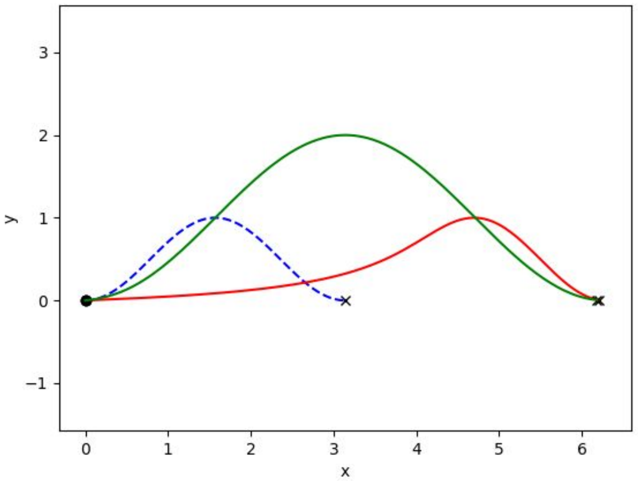
Examples



5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
z
−1.0
−0.5
0.0
0.5
1.0
y
−1.0
−0.5
0.0
0.5
1.0
x



8
6
4
2
0
y
−8
−6
−4
−2
0
2
4
x



3
2
1
0
−1
y
0
1
2
3
4
5
6
x

Obstacles

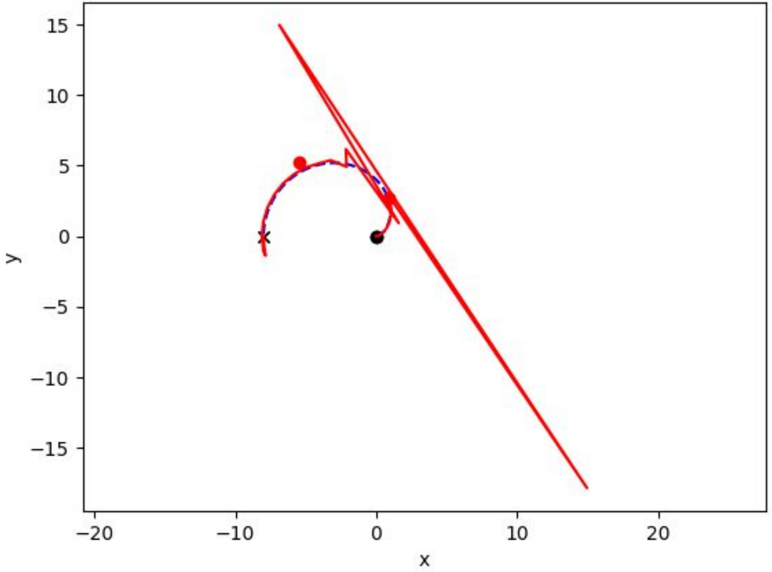
We can include obstacle avoidance by adding to the equations of motion a repellent force, namely a gradient of a potential field centered around the obstacle.

$$\tau\dot{\mathbf{v}} = K(\mathbf{g}-\mathbf{x}) - D\mathbf{v} - K(\mathbf{g}-\mathbf{x}_0) + K\mathbf{f}(s) + \mathbf{p}(\mathbf{x},\mathbf{v})$$

$$\mathbf{p}(\mathbf{x},\mathbf{v}) = -\nabla_{\mathbf{x}} U(\mathbf{x},\mathbf{v})$$

By doing so, we also must handle the fact that too big steps would lead the trajectory inside the potential field, leading to an exploding behaviour.

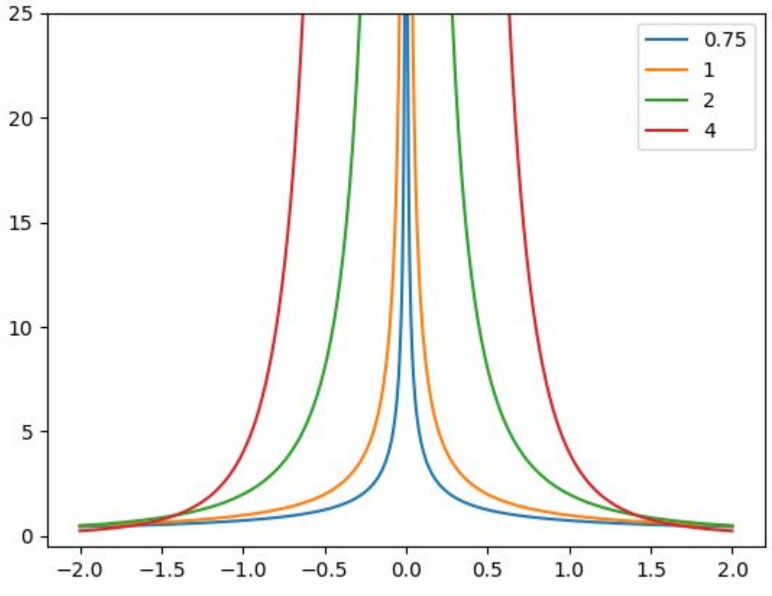
To counter this, an adaptive solver is required; a reliable choice are the Runge-Kutta methods, in particular RK45.

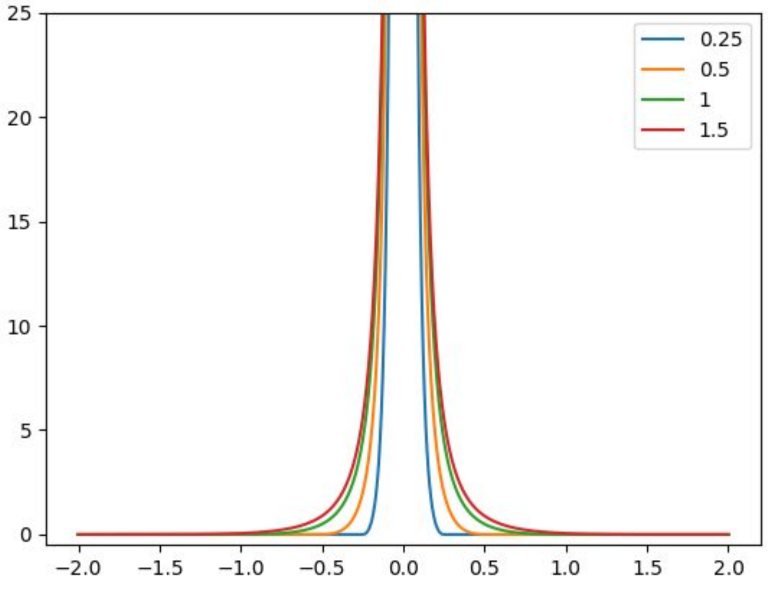


Potentials

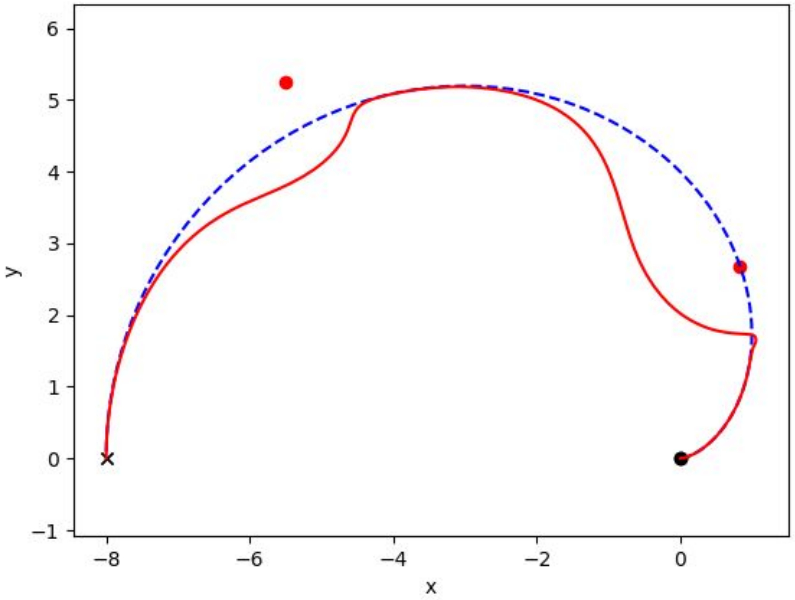
$$U_s(\mathbf{x}) = \frac{1}{r(\mathbf{x})^\beta}$$

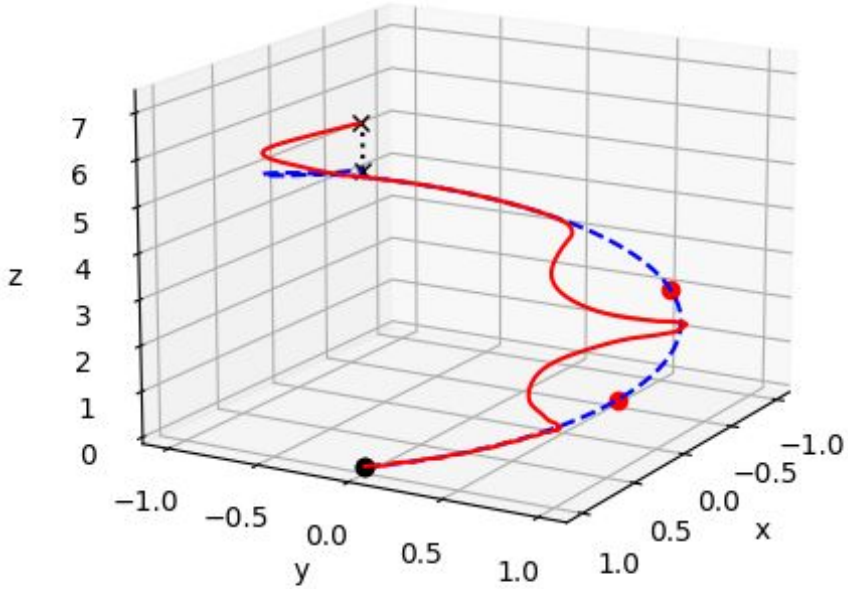
$$U_s(\mathbf{x}) = \begin{cases} \frac{\eta}{2}\left(\frac{1}{r(\mathbf{x})} - \frac{1}{r_0}\right)^2 & \text{if } r(\mathbf{x}) \leq r_0 \\ 0 & \text{otherwise} \end{cases}$$



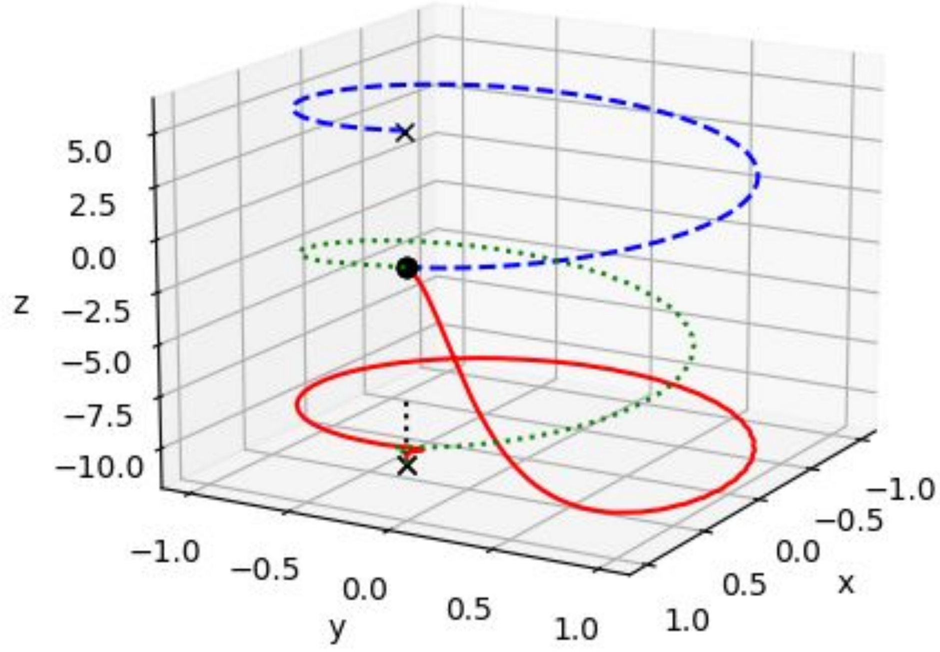


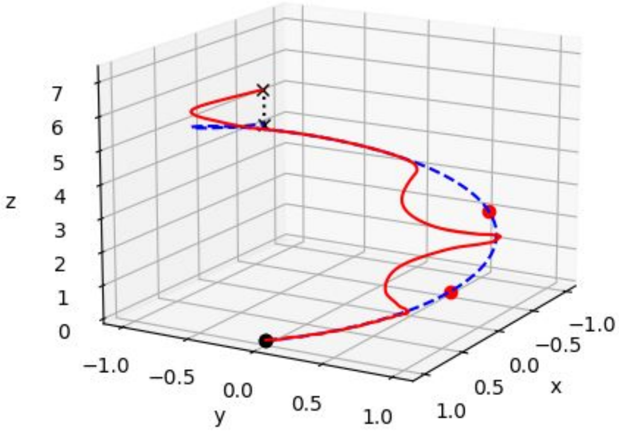
Obstacle avoidance

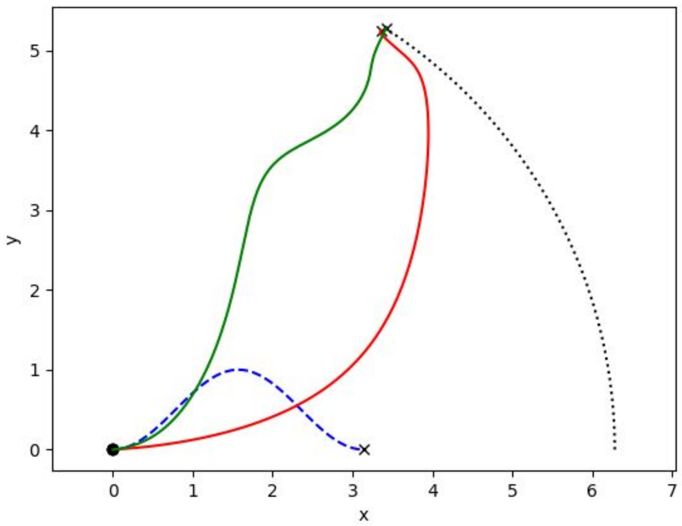




Fancy Examples







Thank you for your attention!