Benchmark examples for AINNCS-2020

Abstract

We present new benchmark examples for the verification of closed-loop dynamical systems with neural network controllers. Because of the complexity of their dynamic systems, the examples offered can be categorized as complex, and perhaps challenging, for any verification system. We believe this will allow a more rigorous comparison be made between various verification systems.

1. Benchmark Description

1.1 Double Pendulum

Our first example includes a two-link pendulum with equal point masses m at the end of each mass-less links of length L. Both links are actuated with torques T_1 and T_2 and we assume viscous friction exists with a coefficient of c. The governing equation of motion can be obtained as:

$$2\ddot{\theta}_1 + \ddot{\theta}_2 \cos(\theta_2 - \theta_1) - \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) - 2\frac{g}{L}\sin\theta_1 + \frac{c}{mL^2}\dot{\theta}_1 = \frac{1}{mL^2}T_1 \tag{1a}$$

$$\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + \ddot{\theta}_2 + \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - \frac{g}{L}\sin\theta_2 + \frac{c}{mL^2}\dot{\theta}_2 = \frac{1}{mL^2}T_2 \tag{1b}$$

where θ_1 and θ_2 are the angles that links make with the upward vertical axis (see fig. 1). The angular velocity and acceleration of links are denoted with $\dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1$ and $\ddot{\theta}_2$ and g is the gravitational acceleration.

First order Euler and second order Runge Kutta explicit time advance scheme are implemented.

1.2 Triple Pendulum

The second example is three-link pendulum, with parameters similar to those introduced for the double pendulum. The equation of motions can be obtained as follows:

$$\begin{aligned} 3\ddot{\theta}_1 + 2\ddot{\theta}_2 \cos(\theta_2 - \theta_1) + \ddot{\theta}_3 \cos(\theta_1 - \theta_3) + 2\dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + \dot{\theta}_3^2 \sin(\theta_1 - \theta_3) \\ -3\frac{g}{L}\sin\theta_1 + \frac{c}{mL^2}\dot{\theta}_1 = \frac{1}{mL^2}T_1 \end{aligned} \tag{2a}$$

$$\begin{aligned} 2\cos(\theta_1 - \theta_2)\ddot{\theta}_1 + 2\ddot{\theta}_2 + \ddot{\theta}_3 \cos(\theta_2 - \theta_3) - 2\dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + \dot{\theta}_3^2 \sin(\theta_2 - \theta_3) \\ -2\frac{g}{L}\sin\theta_2 + \frac{c}{mL^2}\dot{\theta}_2 = \frac{1}{mL^2}T_2 \end{aligned} \tag{2b}$$

$$\begin{aligned} \cos(\theta_1 - \theta_3)\ddot{\theta}_1 + \cos(\theta_2 - \theta_3)\ddot{\theta}_2 + \ddot{\theta}_3 - \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) - \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) \\ -\frac{g}{L}\sin\theta_3 + \frac{c}{mL^2}\dot{\theta}_3 = \frac{1}{mL^2}T_3 \end{aligned} \tag{2c}$$

where θ_1, θ_2 and θ_3 and their time derivatives are defined similar to section 1.1; see fig. 1.

1.3 n-Pendulum

We are prepared to generate an n-link pendulum example for any number of n. Though, we believe this would be too complex for any verification tool.

1.4 Airplane

Airplane example consists of a dynamical system that is a simple model of a flying airplane. The equations of motion are reduced to:

$$\dot{u} = -g\sin\theta + \frac{F_x}{m} - qw + rv \tag{3a}$$

$$\dot{v} = g\cos\theta\sin\phi + \frac{F_y}{m} - ru + pw \tag{3b}$$

$$\dot{w} = g\cos\theta\cos\phi + \frac{F_z}{m} - pv + qu \tag{3c}$$

$$I_x\dot{p} + I_{xz}\dot{r} = M_x - (I_z - I_y)qr - I_{xz}pq \tag{3d}$$

$$I_y\dot{q} = M_y - I_{xz}\left(r^2 - p^2\right) - (I_x - I_z)pr \tag{3e}$$

$$I_{xz}\dot{p} + I_z\dot{r} = M_z - (I_y - I_x)qp - I_{xz}rq \tag{3f}$$

where here (u, v, w) are the components of velocity in (x, y, z) directions, (p, q, r) are blah blah blah and (ϕ, θ, ψ) are the Euler angles. The mass of the airplane is denoted with m and I_x, I_y, I_z and I_{xz} are the moment of inertia with respect to the indicated axis; see fig. 1. The controls parameters include three force components F_x, F_y and F_z and three moment components M_x, M_y, M_z. Notice that for simplicity we have assumed the aerodynamic forces are absorbed in the F's. In addition to these six equations, we have six additional kinematic equations:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix} \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi & \cos\phi\end{bmatrix} \begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{4}$$

and

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} = \begin{bmatrix}1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sec\theta\sin\phi & \sec\theta\cos\phi\end{bmatrix} \begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{5}$$

For the simplicity of control design, here we have chosen parameter to have some nominal dimensionless values: $m = 1, I_x = I_y = I_z = 1, I_{xz} = 0$ and $g = 1$.

2. Training Neural Networks

We have provided a general library for training controller for either of the benchmarks. The training uses *behavior cloning*, a supervised learning approach for learning controllers. Here a neural network is trained to replicate an expert demonstrations. More specifically, we initially generate a set of *expert* control parameters for different initial states of the system. *Expert* control parameters are defined as those that lead the system to reach to its goal state in finite time.

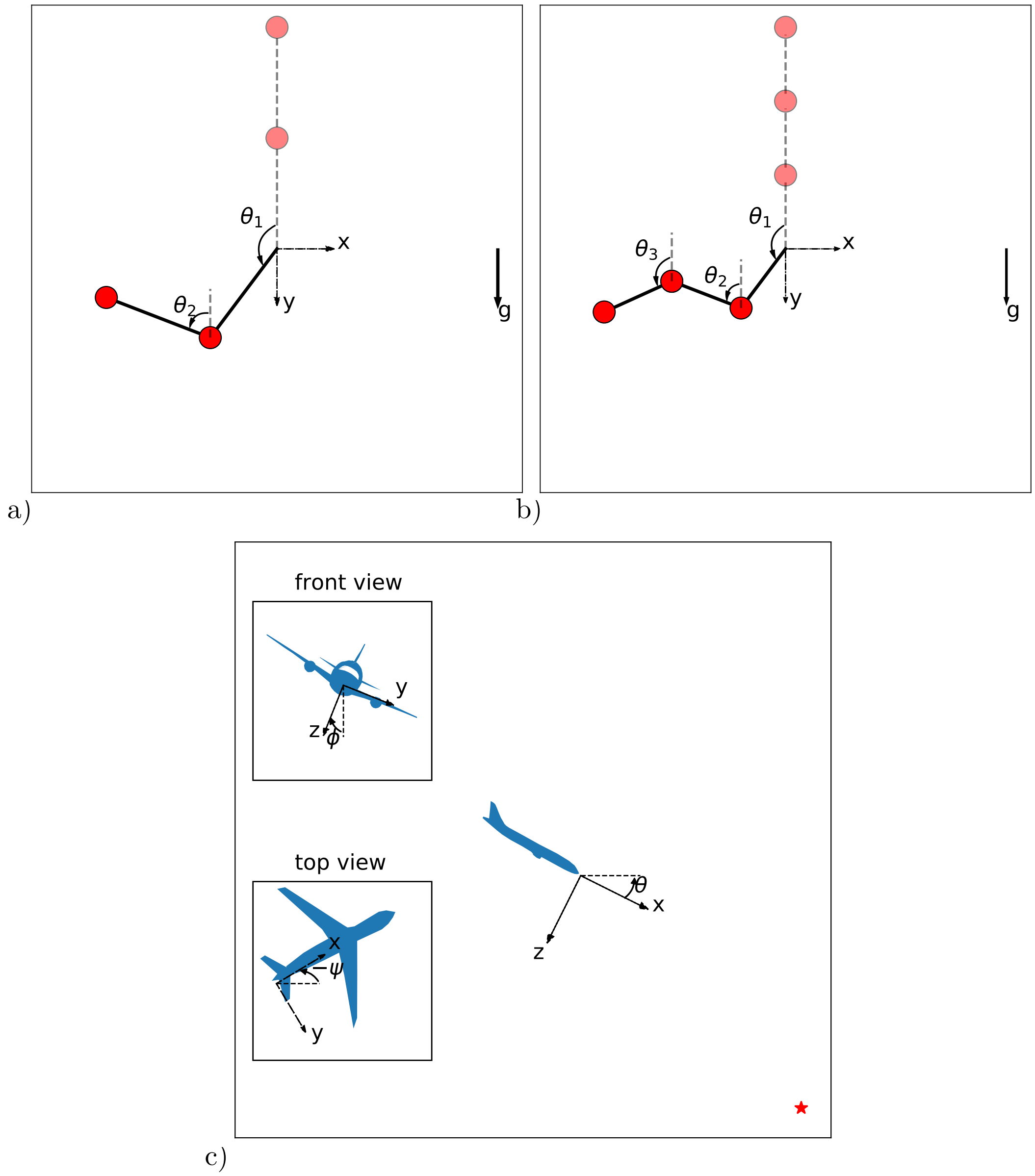

Figure 1: Control problem examples: a) inverted double pendulum. The goal is keep the pendulum upright (dashed schematics); b) inverted triple pendulum. The goal is keep the pendulum upright (dashed schematics); c) airplane. The goal is to reach to the star, with no lateral change in position and and all Euler angles equal to 0

The expert control parameters are generated using technique in optimal control. Specifically, we have provided an implementation of LQR (Linear Quadratic Regulator) and iLQR(iterative LQR). Users can also use other algorithms such as trajectory optimization (ref).

For the double pendulum problem, we have used a mixture of iLQR (iterative Linear Quadratic Regulator) and LQR (Linear Quadratic Regulator) to generate the data. Initially, the control parameters are provided by iLQR, and once the pendulum is near the goal state, LQR will take over and stabilize the pendulum. In our data generation procedure, the double pendulum is to be stabilized within 400 time steps (with $\Delta t = 0.01$). The first 150 time steps are controlled by the iLQR controller and the rest is controlled by the LQR controller. During training, we have trained two separate neural networks for iLQR and LQR generate data. Both networks are identical in configuration with three hidden layers of size x by y by z. The input and out layer sizes, dictated by the problem, are 4 (=number of states) and 2 (=number of control parameters), respectively. For training, we have used a Mean Square Error loss function and Adam optimizer with an initial learning rate of 0.01 that gradually reduces to 0.0001, as training progresses.