Benchmark examples for AINNCS-2020

Amir Maleki
amir.maleki@stanford.edu

Chelsea Sidrane
csidrane@stanford.edu

June 17, 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 Single Pendulum

This is the classical inverted pendulum environment. A ball of mass m is attached to a massless beam of length L. The beam is actuated with a torque T and we assume viscous friction exists with a coefficient of c. The governing equation of motion can be obtained as:

$$\ddot{\theta} = \frac{g}{L}\sin\theta + \frac{1}{mL^2}\left(T - c\,\dot{\theta}\right) \tag{1}$$

where θ is the angle that link makes with the upward vertical axis, and $\dot{\theta}$ is the angular velocity. The state vector is:

$$[\theta, \dot{\theta}] \tag{2}$$

1.2 Double Pendulum

Our next example includes an inverted two-link pendulum with equal point masses m at the end of connected 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:

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

$$\ddot{\theta}_1 \cos(\theta_2 - \theta_1) + \ddot{\theta}_2 + \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) - \frac{g}{L}\sin\theta_2 + \frac{c}{mL^2}\dot{\theta}_2 = \frac{1}{mL^2}T_2 \tag{3b}$$

where θ_1 and θ_2 are the angles that links make with the upward vertical axis (see fig. 1). The state is:

$$[\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2] \tag{4}$$

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.

1.3 Triple Pendulum

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

$$3\ddot{\theta}_1 + 2\ddot{\theta}_2 \cos(\theta_2 - \theta_1) + \ddot{\theta}_3 \cos(\theta_1 - \theta_3) + 2\dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + \dot{\theta}_3^2 \sin(\theta_1 - \theta_3) - 3\frac{g}{L}\sin\theta_1 + \frac{c}{mL^2}\dot{\theta}_1 = \frac{1}{mL^2}T_1 \tag{5a}$$

$$2\cos(\theta_1 - \theta_2)\ddot{\theta}_1 + 2\ddot{\theta}_2 + \ddot{\theta}_3 \cos(\theta_2 - \theta_3) - 2\dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + \dot{\theta}_3^2 \sin(\theta_2 - \theta_3) - 2\frac{g}{L}\sin\theta_2 + \frac{c}{mL^2}\dot{\theta}_2 = \frac{1}{mL^2}T_2 \tag{5b}$$

$$\cos(\theta_1 - \theta_3)\ddot{\theta}_1 + \cos(\theta_2 - \theta_3)\ddot{\theta}_2 + \ddot{\theta}_3 - \dot{\theta}_1^2 \sin(\theta_1 - \theta_3) - \dot{\theta}_2^2 \sin(\theta_2 - \theta_3) - \frac{g}{L}\sin\theta_3 + \frac{c}{mL^2}\dot{\theta}_3 = \frac{1}{mL^2}T_3 \tag{5c}$$

where θ_1, θ_2 and θ_3 and their time derivatives are defined similar to section 1.2; see fig. 1.

1.4 n-Pendulum

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

1.5 Airplane

Airplane example consists of a dynamical system that is a simple model of a flying airplane. The state is:

$$[x, y, z, u, v, w, \phi, \theta, \psi, r, p, q] \tag{6}$$

where (x, y, z) is the position of the C.G., (u, v, w) are the components of velocity in (x, y, z) directions, (p, q, r) are body rotation rates, and (ϕ, θ, ψ) are the Euler angles. The equations of motion are reduced to:

$$\dot{u} = -g\sin\theta + \frac{F_x}{m} - qw + rv \tag{7a}$$

$$\dot{v} = g\cos\theta\sin\phi + \frac{F_y}{m} - ru + pw \tag{7b}$$

$$\dot{w} = g\cos\theta\cos\phi + \frac{F_z}{m} - pv + qu \tag{7c}$$

$$I_x\dot{p} + I_{xz}\dot{r} = M_x - (I_z - I_y)qr - I_{xz}pq \tag{7d}$$

$$I_y\dot{q} = M_y - I_{xz}\left(r^2 - p^2\right) - (I_x - I_z)pr \tag{7e}$$

$$I_{xz}\dot{p} + I_z\dot{r} = M_z - (I_y - I_x)qp - I_{xz}rq \tag{7f}$$

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:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{8}$$

and

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} = \begin{bmatrix}1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sec\theta\sin\phi & \sec\theta\cos\phi\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{9}$$

2 Neural Network Controllers

In addition to the environments, we have provided some neural network controllers. These controllers are trained using *behavior cloning*, a supervised

learning approach for training controllers. Here, a neural network is trained to replicate expert demonstrations. We initially generate a set of *expert* control inputs for trajectories originating from different initial states of the system. *Expert* control inputs are defined as those that lead the system to reach to its goal state in finite time.

The expert control inputs are generated using optimal control techniques. Specifically, we have used an implementation of LQR (Linear Quadratic Regulator) and iLQR(iterative LQR) control. The codes for these implementation, as well training procedure are provided.

For the double pendulum problem, we have used a mixture of iLQR and LQR 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, one from the iLQR data and one from the LQR data. Both networks are identical in configuration with three hidden layers of size x by y by z. The input and output 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 squared error loss function and the Adam optimizer with an initial learning rate of 0.01 that gradually reduces to 0.0001, as training progresses.

3 Benchmarks specifications

For the purpose of the benchmark, we have decided to work on controllers that are trained with iLQR-generated data. This is a completely arbitrary choice, and one may define similar specifications for a network with LQR-generated data. The following specifications are aimed to focus around areas that the controllers are more robust to create more interesting safety problems. The controller files are provided in nnet, onnx and h5 (keras model) formats. We suggest the following safety specifications to be used with these benchmark examples:

- **Single Pendulum** Model parameters are

$$m = 0.5, L = 0.5, c = 0., g = 1.0$$

  Controller is:

1. `controller single pendulum`: Use this controller with $\Delta t = 0.05$. The initial set is

$$[\theta, \dot{\theta}] = [1.0, 1.2] \times [0.0, 0.2].$$

Safety specification is

$$\forall n_t : 10 \leq n_t \leq 20, \theta \in [0.0, 1.0].$$

- **Double Pendulum** Model parameters are

$$m = 0.5, L = 0.5, c = 0., g = 1.0$$

Controllers are:

1. `controller double pendulum less robust`: Use this controller with $\Delta t = 0.05$. The initial set is

$$[\theta_1, \theta_2, \dot{\theta}_1\dot{\theta}_2] = [1.0, 1.3]^4.$$

Safety specification is

$$\forall n_t \leq 20, [\theta_1, \theta_2, \dot{\theta}_1\dot{\theta}_2] \in [-1.0, 1.7]^4.$$

2. `controller double pendulum more robust`: Use this controller with $\Delta t = 0.02$. The initial set is

$$[\theta_1, \theta_2, \dot{\theta}_1\dot{\theta}_2] = [1.0, 1.3]^4.$$

Safety specification is

$$\forall n_t \leq 20, [\theta_1, \theta_2, \dot{\theta}_1\dot{\theta}_2] \in [-0.5, 1.5]^4.$$

- **Triple Pendulum** Model parameters are

$$m = 0.5, L = 0.5, c = 0., g = 1.0$$

Controller is:

1. `controller tripel pendulum`: Use this controller with $\Delta t = 0.02$. The initial set is

$$[\theta_1, \theta_2, \theta_3] = [0.5, 0.7]^3, [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3] = [-0.5, -0.3]^3.$$

Safety specification is

$$\forall n_t : n_t \leq 20, [\theta_1, \theta_2] \in [0.0, 0.7]^2.$$

- **Airplane** Model parameters are

$$m = 1, I_x = I_y = I_z = 1, I_{xz} = 0, g = 1$$

1. `controller airplane`: Use this controller with $\Delta t = 0.1$. The initial set is

$$x = y = z = r = p = q = 0, [u, v, w, \phi, \theta, \psi] = [0.0, 1.0]^6.$$

Safety specification is

$$\forall n_t : n_t \leq 20, y \in [-0.5, 0.5], [\phi, \theta, \psi] = [-1.0, 1.0]^3.$$

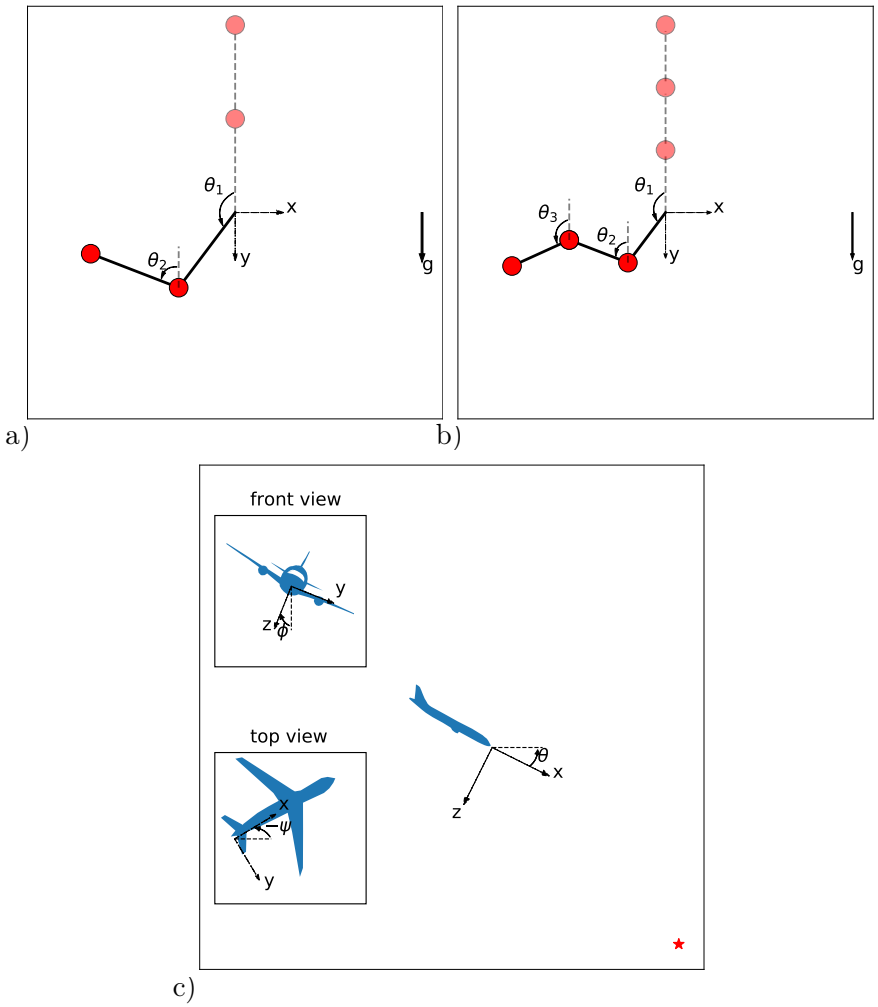

Figure 1: Control problem examples: a) inverted double pendulum. The goal is keep the pendulum upright (dashed schematics); b) inverted triple pendulum. The goal is keep the pendulum upright (dashed schematics); c) airplane.