Learning Bicycle Stunts:
Supplementary Document

Jie Tan Yuting Gu C. Karen Liu Greg Turk

Georgia Institute of Technology

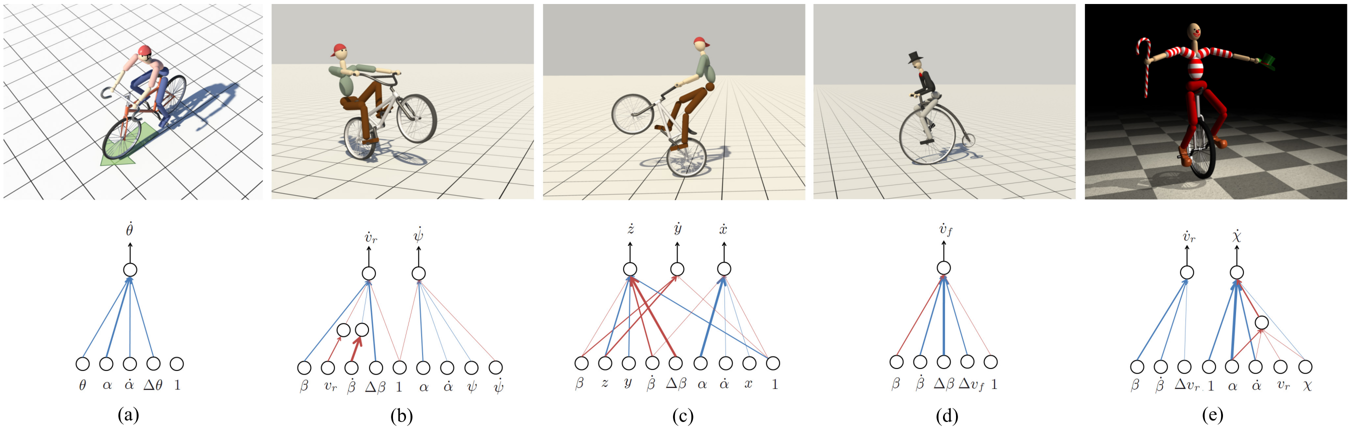

Figure 1: *Balance-driven tasks and the neural network structures of their corresponding controllers: (a) balance and steering, (b) wheelie, (c) back hop, (d) riding a high wheeler (stunt) and (e) riding a unicycle. Red edges have positive weights and blue edges have negative weights. The thickness of the edges shows the relative magnitudes of the weights.*

1 Neural Network Structures

We used NEAT to search for both the topologies and the weights of neural networks for balance-driven tasks. In addition to the endo balance controller, which is shown in the paper, we illustrate other learned neural networks in Figure 1. Recall that NEAT begins each controller with direct connections between input and output neurons. Note that NEAT performs feature selection (removing connections) for the controllers of balance and steering and back hop. It also performs structural complexification (adding nodes and connections) in the examples of the wheelie and riding a unicycle. We find that it is difficult to interpret these network structures, and this is a common problem of using neural networks.

2 Reward Functions

We summarize the reward functions that are used to learn different bicycle tasks in Table 1. We accumulate the rewards for 1000 time steps or until the bicycle loses its balance $|\alpha| > 0.5$ or when the stunt fails $|\Delta\beta| > 0.5$.

Task	Reward Function
momentum-driven	
going over curbs	β
endo (lifting)	$-\beta$
front wheel pivot	$\begin{cases} 500\dot{\gamma}\Delta t & \text{if the pivoting phase has started,} \\ 1 & \text{if the pivoting phase has ended,} \\ 0 & \text{otherwise.} \end{cases}$
bunny hop	$h_f h_r$
balance-driven	
balance and steering	$1 + \frac{1}{\Delta\theta+1}$
wheelie	$1 + \frac{1}{\Delta\beta+1} + \frac{1}{\psi+0.1} + \frac{2}{\alpha+0.1}$
endo (balance)	$1 + \frac{1}{\Delta\beta+1} + \frac{1}{\Delta v_f+0.1} + \frac{1}{\alpha+1} + \frac{1}{\theta+1}$
back hop	$1 + \frac{1}{y+0.1} + \frac{1}{z+0.1} + \frac{1}{x+0.1}$
high wheeler (stunt)	$1 + \frac{1}{\Delta\beta+1} + \frac{1}{\Delta v_f+1}$
unicycle	$1 + \frac{1}{\Delta v_r+1} + \frac{2}{\alpha+1} + \frac{1}{\chi+1}$

Table 1: *Reward functions for different tasks.*

3 Bicycle Specifications

We describe the physical specifications of the three bicycles and the unicycle in Table 2.

Specification	Road Bike	BMX Bike	High Wheeler	Unicycle
mass of the frame	7.0	5.6	8.0	2.0
mass of the handlebar	3.5	3.5	4.0	NA
mass of the front wheel	1.7	0.71	5.0	NA
mass of the rear wheel	1.7	0.71	0.5	2.3
radius of the front wheel	0.32	0.25	0.45	NA
radius of the rear wheel	0.32	0.25	0.11	0.32
width of the tires	0.015	0.03	0.015	0.03
distance between wheels	1.03	0.85	0.58	NA

Table 2: *Specifications of different bicycles and the unicycle. The mass unit is kg and the length unit is m. The distance between wheels only accounts for the horizontal distance.*

4 Simulation and Optimization Parameters

We used Open Dynamic Engine as our physical simulator. Table 3 summarizes the simulation parameters used in our examples.

We applied CMA to search for the splines for the momentum-driven tasks. We used 90 samples per iteration, a maximum of 50 iterations and the default values for other parameters [Hansen 2006].

Parameter	Value
time step	0.01s
constraint force mixing (CFM)	10^{-10}
error reduction parameter (ERP)	0.99
friction coefficient (BMX bike examples)	2.0
friction coefficient (other examples)	1.0

Table 3: *Simulation parameters.*

Parameter	Value
max number of iterations	50
population size	90
survival rate	0.2
crossover rate	0.7
mutation rate	0.2
chance of adding a link	0.05
chance of adding a node	0.05
chance of replacing a weight	0.1
max weight perturbation	0.5

Table 4: *NEAT parameters.*

We applied NEAT to search for the neural networks for the balance-driven tasks. The implementation can be found at Stanley [2002]. Table 4 summarizes the parameters used in our NEAT optimization.

5 Constraints in ODE

We are going to describe how various constraints are formulated in ODE for completeness of presentation.

Joint constraints Joints that connect two rigid bodies constrain their relative motions. The hinge, universal and ball joints impose five, four and three constraints respectively, each of which is a linear equality constraint.

$$\mathbf{J}_A \begin{bmatrix} \mathbf{v}_A \\ \omega_A \end{bmatrix} - \mathbf{J}_B \begin{bmatrix} \mathbf{v}_B \\ \omega_B \end{bmatrix} = 0 \tag{1}$$

where $\mathbf{J}_A$ is a row in the Jacobian matrix that maps the velocities of body A to one component of the linear velocity at the joint position or the angular velocity perpendicular to the joint axes.

Actuator constraints An actuator is attached to each joint of the human character to enable it to actively control its joint motion. The actuators generate internal torques to track the desired pose given by the IK solver (See Figure 2 in the paper). The following linear equality constraint should be satisfied for each actuated degree of freedom (DOF).

$$\mathbf{n}^T(\omega_A - \omega_B) - \tilde{\dot{q}} = 0 \tag{2}$$

where $\mathbf{n}$ is the axis of the actuated DOF. $\tilde{\dot{q}}$ is the desired actuator angular speed, which is the difference between the desired and current DOF value, divided by the time step.

Contact constraints ODE uses the standard friction pyramid to model the contact forces between the bicycle and the ground. Let $f_\perp$, $f_\parallel^1$ and $f_\parallel^2$ be the normal and two tangential components of the contact force $\mathbf{f}_c$.

$$\mathbf{f}_c = f_\perp \mathbf{n} + f_\parallel^1 \mathbf{t}_1 + f_\parallel^2 \mathbf{t}_2$$

where $\mathbf{n}$ is the ground normal, $\mathbf{t}_1$ and $\mathbf{t}_2$ are the two orthogonal tangential bases of the ground.

Along the normal direction, the contact velocity and contact force satisfy the following linear complementarity constraints.

$$v_\perp \geq 0,\ f_\perp \geq 0,\ v_\perp f_\perp = 0 \tag{3}$$

The normal contact velocity $v_\perp$ can be calculated by multiplying the Jacobian at the contact location with the body velocities and then projected along the normal direction.

$$v_\perp = \mathbf{n}^T \mathbf{J} \begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} \tag{4}$$

Along the tangential direction, one of the following friction cone conditions must be satisfied.

$$\begin{array}{ll} v_\parallel^i > 0, & f_\parallel^i = -\mu f_\perp \\ v_\parallel^i < 0, & f_\parallel^i = \mu f_\perp \\ v_\parallel^i = 0, & -\mu f_\perp \leq f_\parallel^i \leq \mu f_\perp \end{array} \tag{5}$$

where $i \in \{1, 2\}$ and μ is the friction coefficient. The tangential velocities $v_\parallel^i$ can be calculated similar to eq. (4).

The dynamics equation, together with all the constraints (1), (2), (3) and (5), form a mixed linear complementarity program, which can be efficiently solved using a variant of Dantzig's algorithm [Smith 2008].

References

HANSEN, N., 2006. Covariance matrix adaptation: source code. https://www.lri.fr/~hansen/cmaes_inmatlab.html.

SMITH, R., 2008. Open dynamics engine. http://www.ode.org/.

STANLEY, K., 2002. Neuroevolution of augmenting topology: source code. http://www.cs.ucf.edu/~kstanley/neat.html.