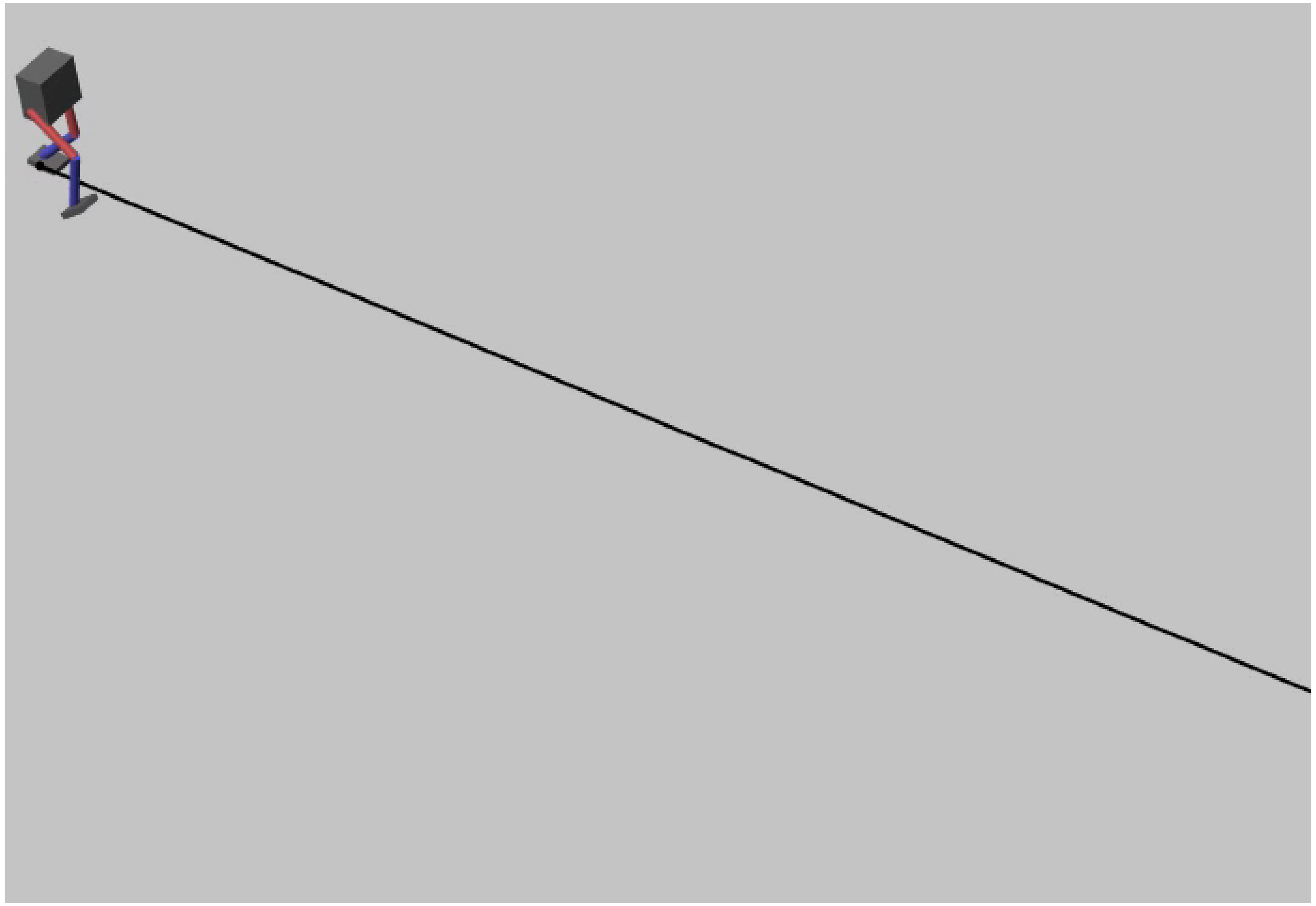
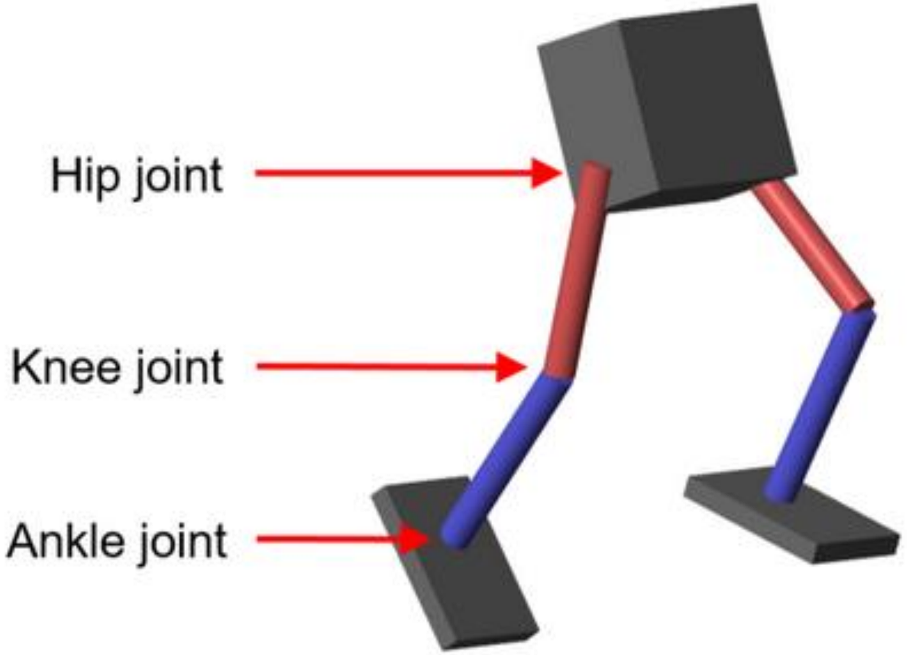
Reinforcement Learning Workflows for AI

Key Takeaways

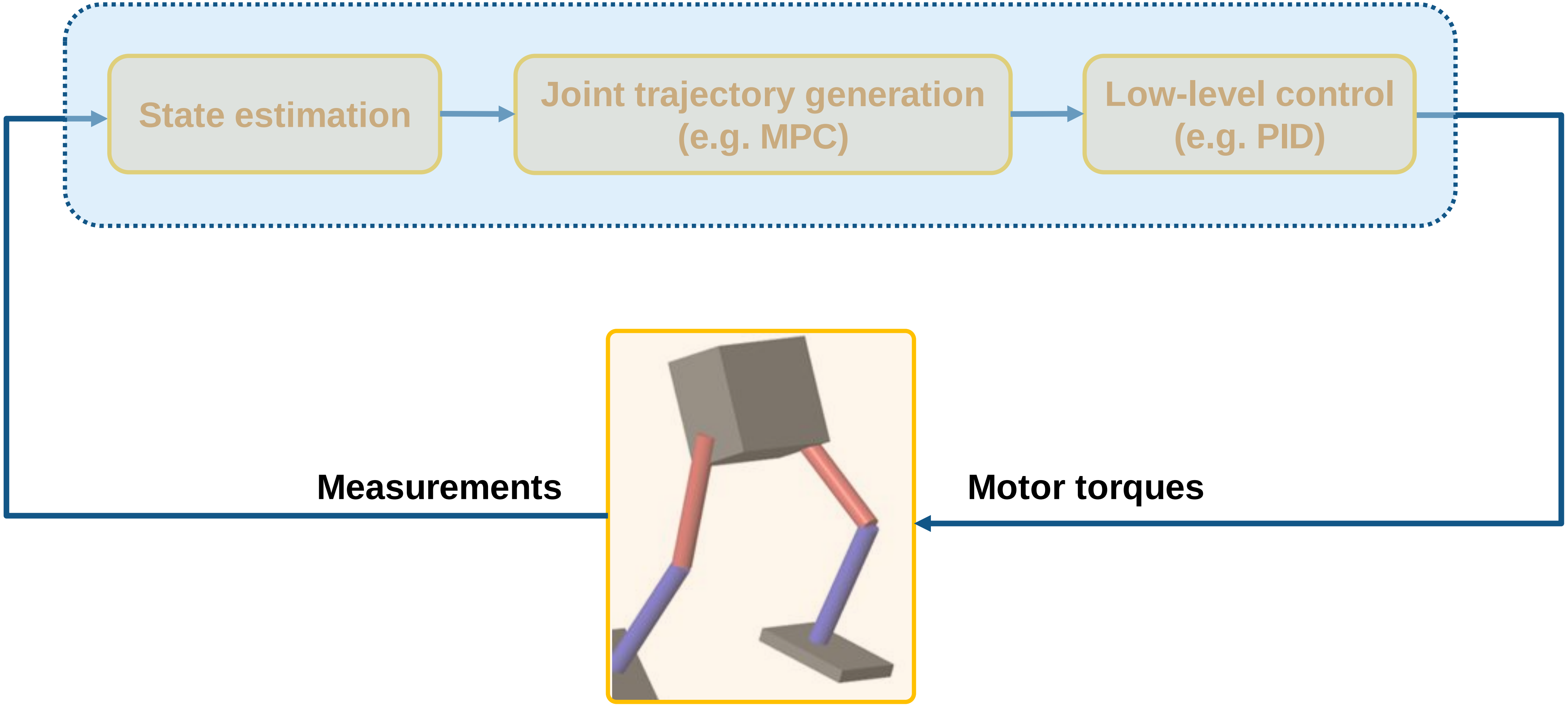
- What is reinforcement learning and why should I care about it?
- How do I set up and solve a reinforcement learning problem?
- What are some common challenges?

Why Should You Care About Reinforcement Learning?

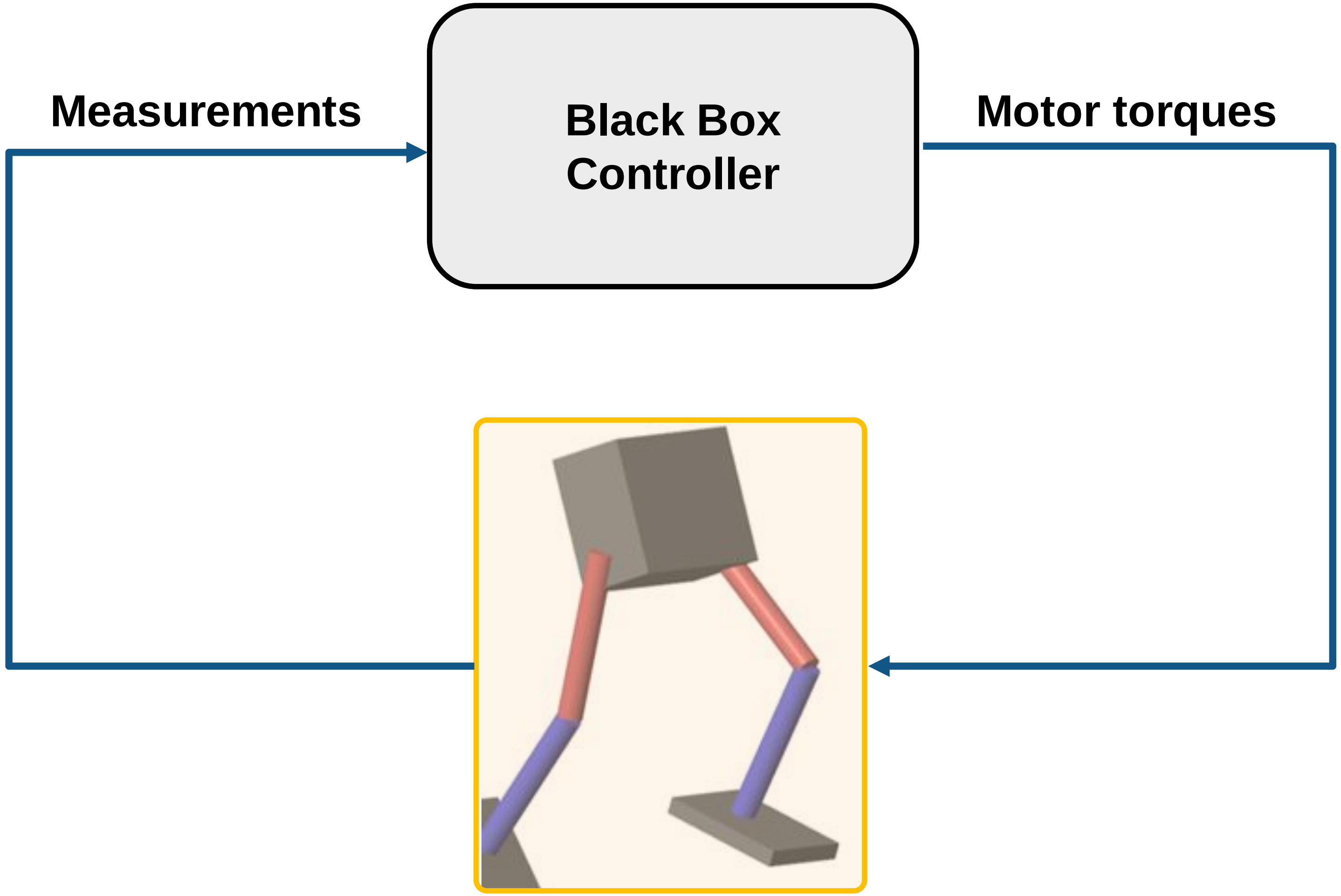




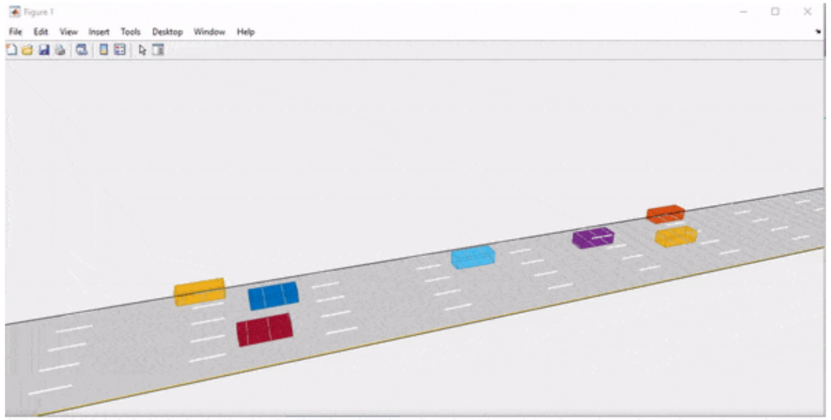
One Approach Could Be…

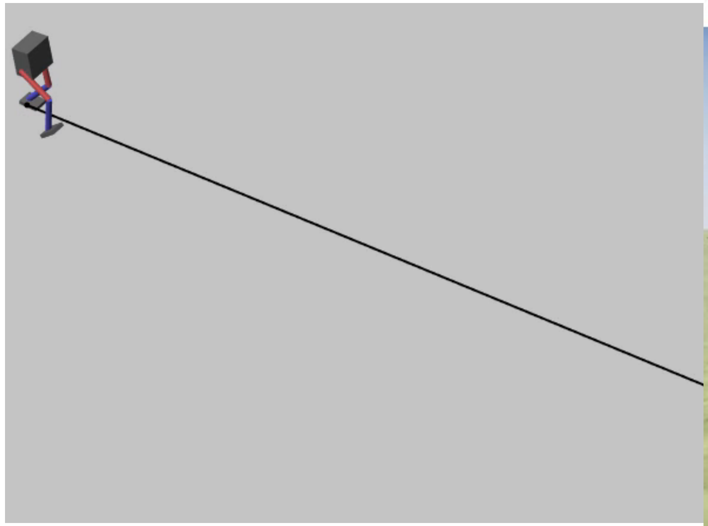


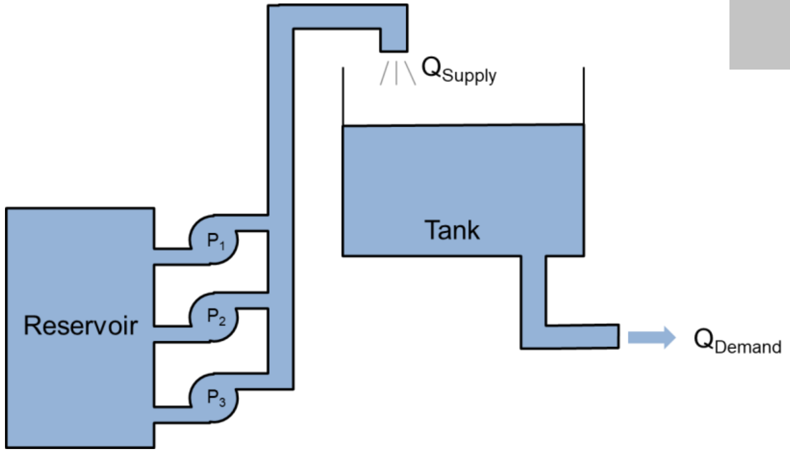
Any Alternatives?

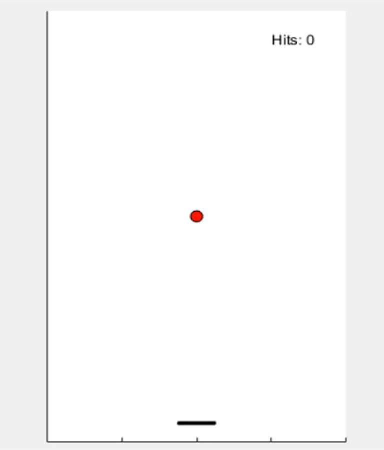


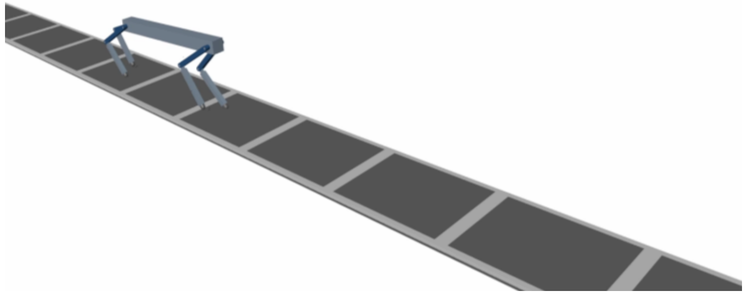
Applications of Reinforcement Learning











What is reinforcement learning?

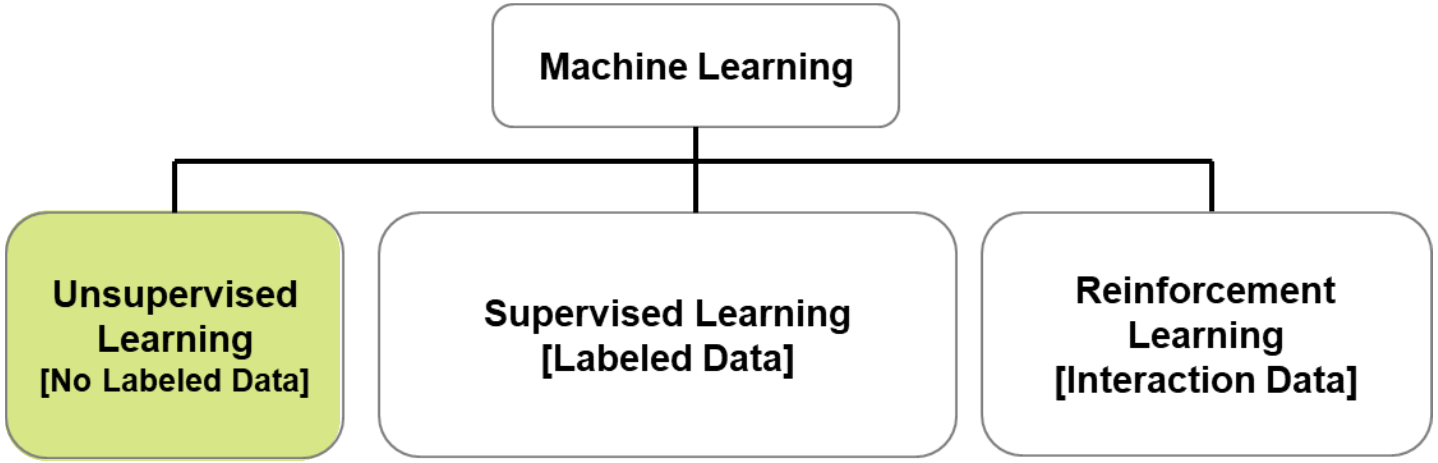
Type of machine learning that trains an **‘agent’** through trial & error interactions with an **environment**

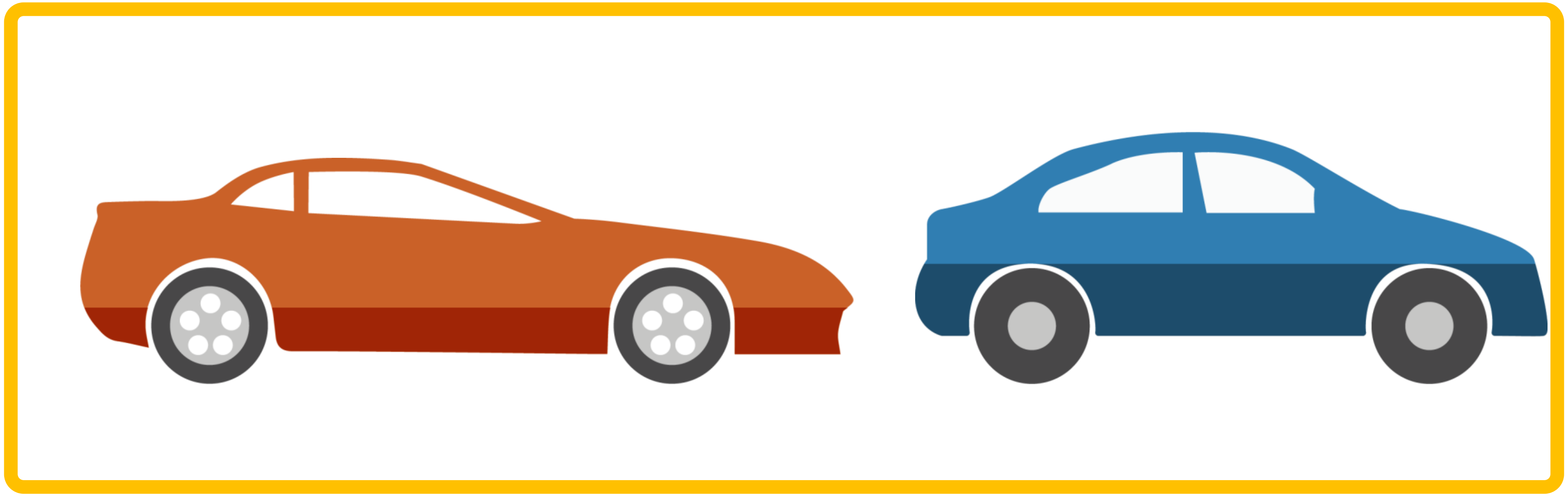
Reinforcement Learning vs Machine Learning vs Deep Learning

Machine Learning

- Unsupervised Learning
  [No Labeled Data]
- Supervised Learning
  [Labeled Data]
- Reinforcement Learning
  [Interaction Data]

Reinforcement Learning vs Machine Learning vs Deep Learning





Reinforcement Learning vs Machine Learning vs Deep Learning

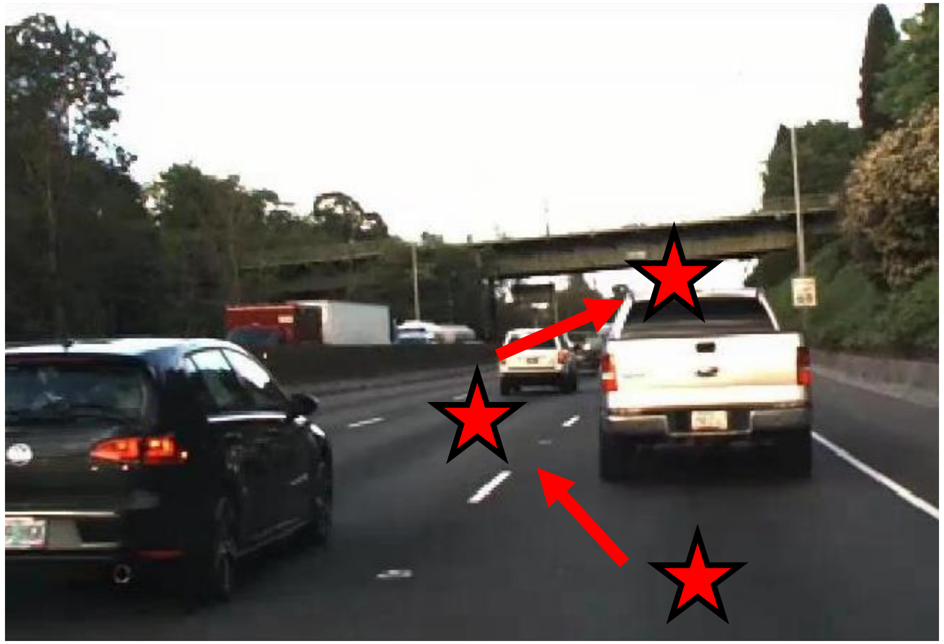
Machine Learning

- Unsupervised Learning [No Labeled Data]
- Supervised Learning [Labeled Data]
- Reinforcement Learning [Interaction Data]

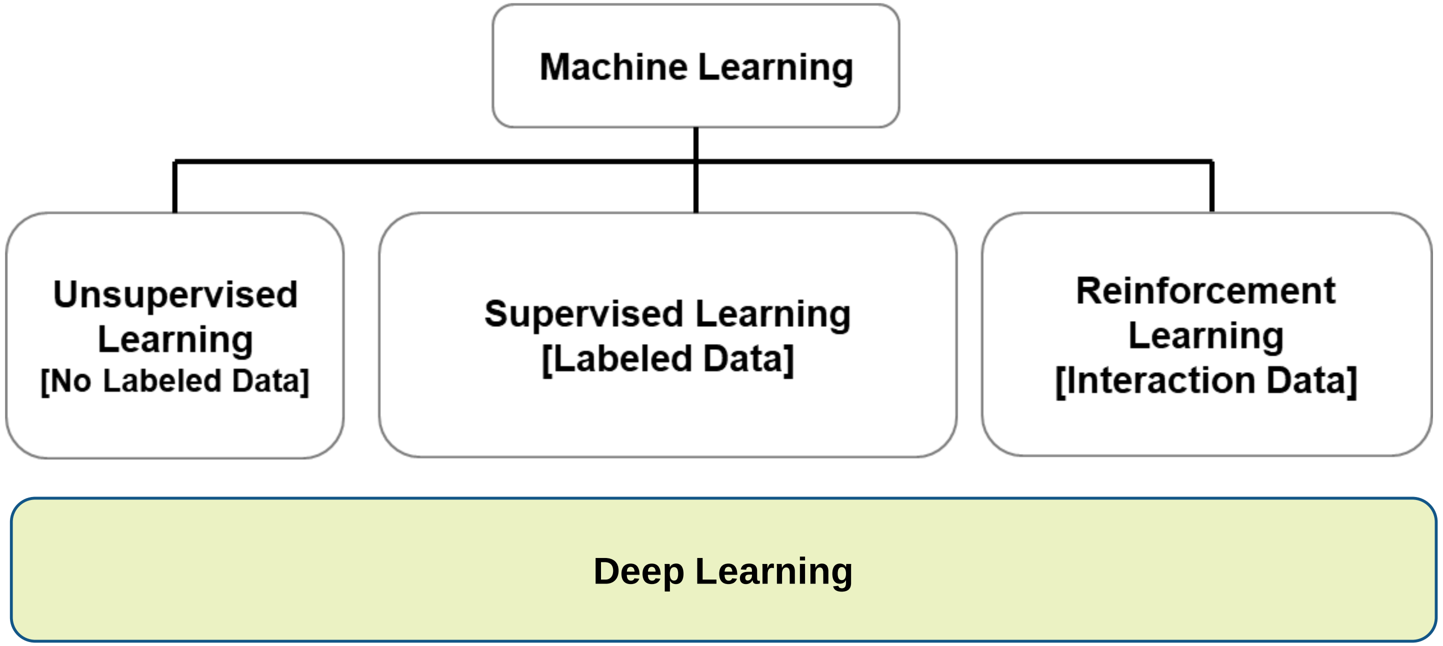
Reinforcement Learning vs Machine Learning vs Deep Learning

Machine Learning

- **Unsupervised Learning** [No Labeled Data]
- **Supervised Learning** [Labeled Data]
- **Reinforcement Learning** [Interaction Data]



Reinforcement Learning vs Machine Learning vs Deep Learning

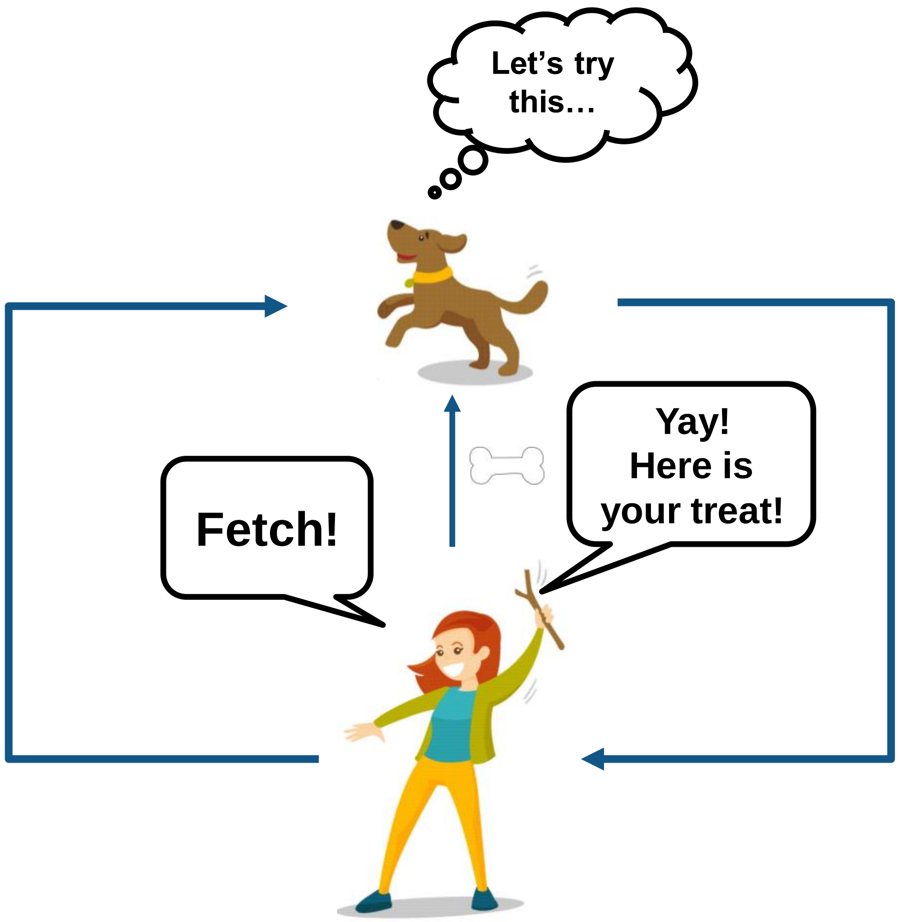


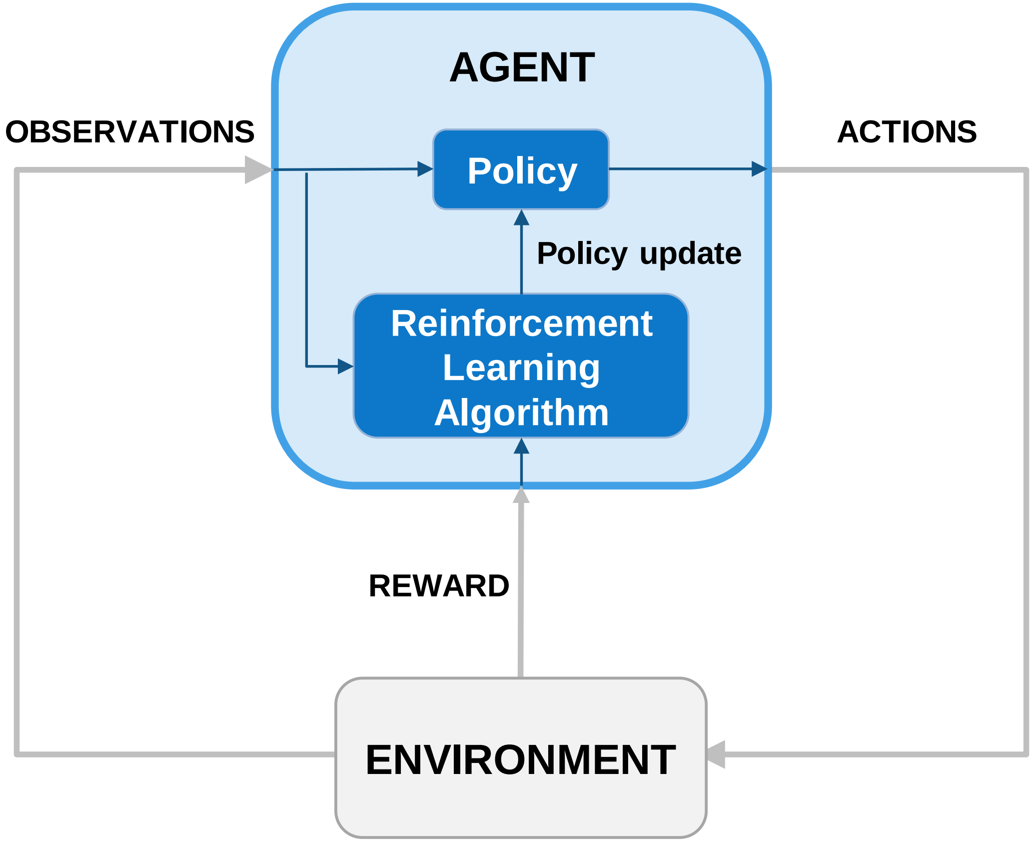
What about deep learning?

Complex reinforcement learning problems typically need deep neural networks
[Deep Reinforcement Learning]

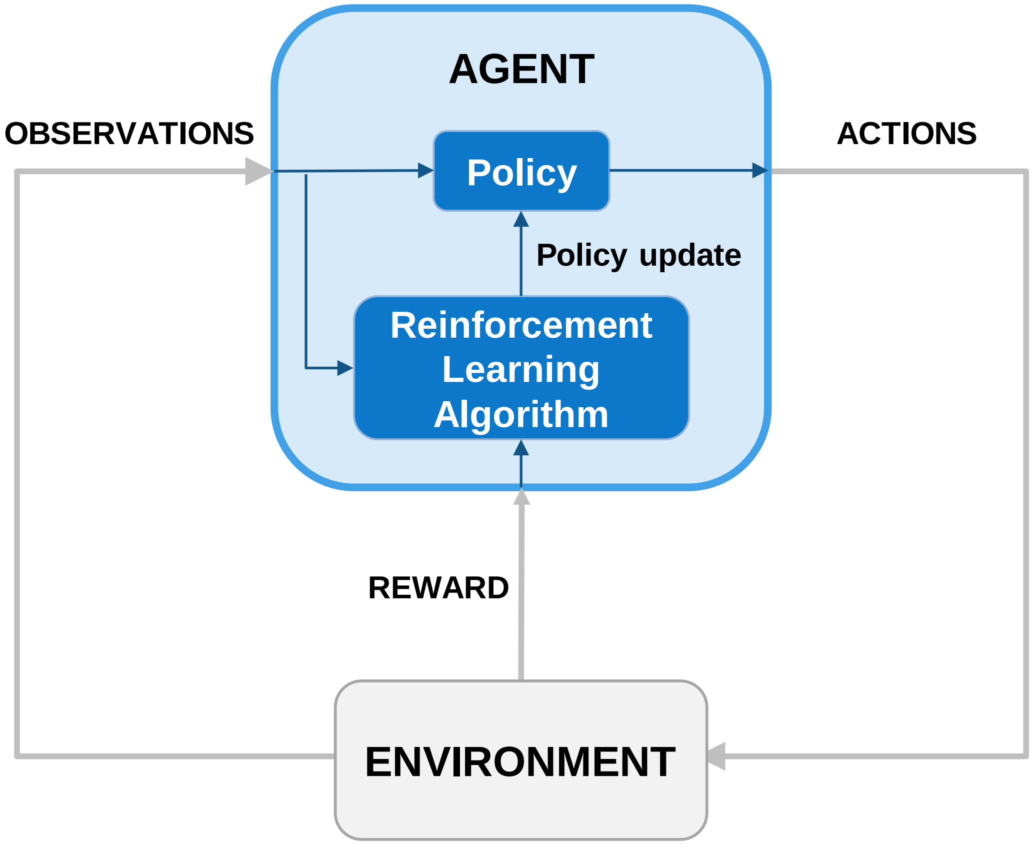
How does reinforcement learning training work?

Analogies with pet training





Reinforcement Learning Concepts
Training a self-driving car

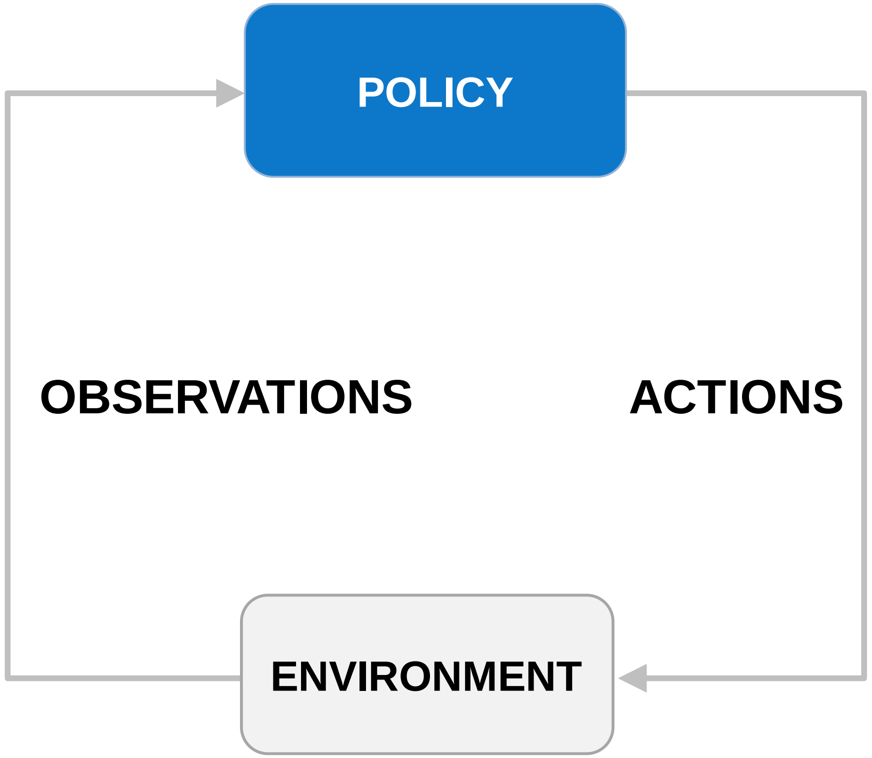


- Vehicle's computer…
  (**agent**)
- is reading sensor measurements from LIDAR, cameras,…
  (**observations**)
- that represent road conditions, vehicle position,…
  (**environment**)
- and generates steering, braking, throttle commands,…
  (**action**)
- based on an internal state-to-action mapping…
  (**policy**)
- that tries to optimize, e.g., lap time & fuel efficiency…
  (**reward**).

- The policy is updated through repeated trial-and-error by a **reinforcement learning algorithm**

Reinforcement Learning Concepts
Training a self-driving car

After training, only trained policy is needed

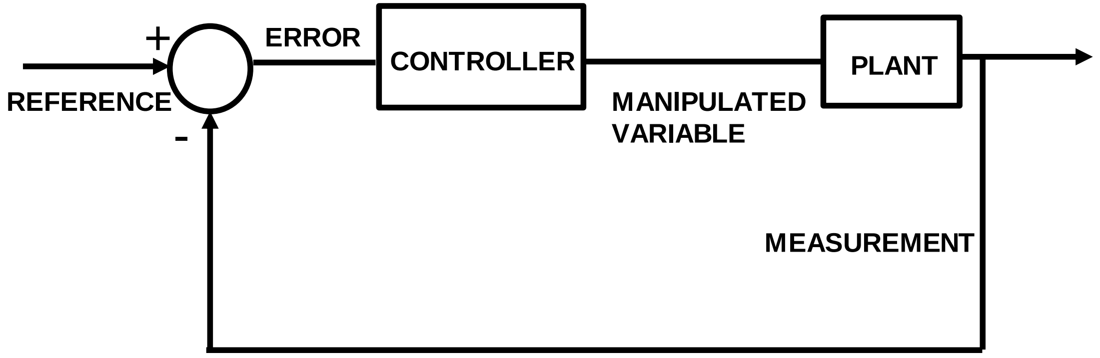


- Vehicle's computer uses the final state-to-action mapping… (**policy**)
- to generate steering, braking, throttle commands,… (**action**)
- based on sensor readings from LIDAR, cameras,… (**observations**)
- that represent road conditions, vehicle position,… (**environment**).

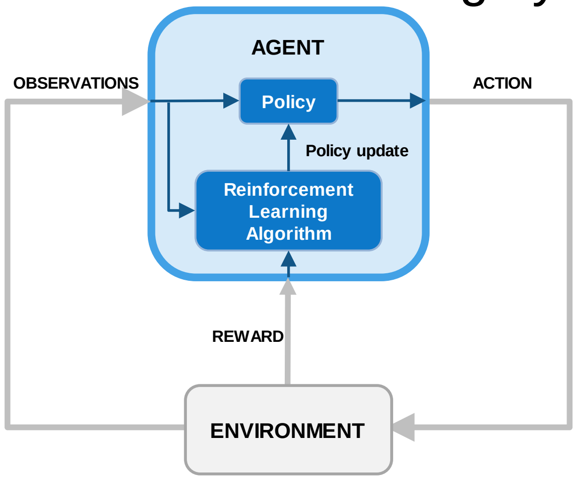
By definition, this trained policy is optimizing lap time & fuel efficiency

Reinforcement Learning vs Controls

Control system



Reinforcement learning system



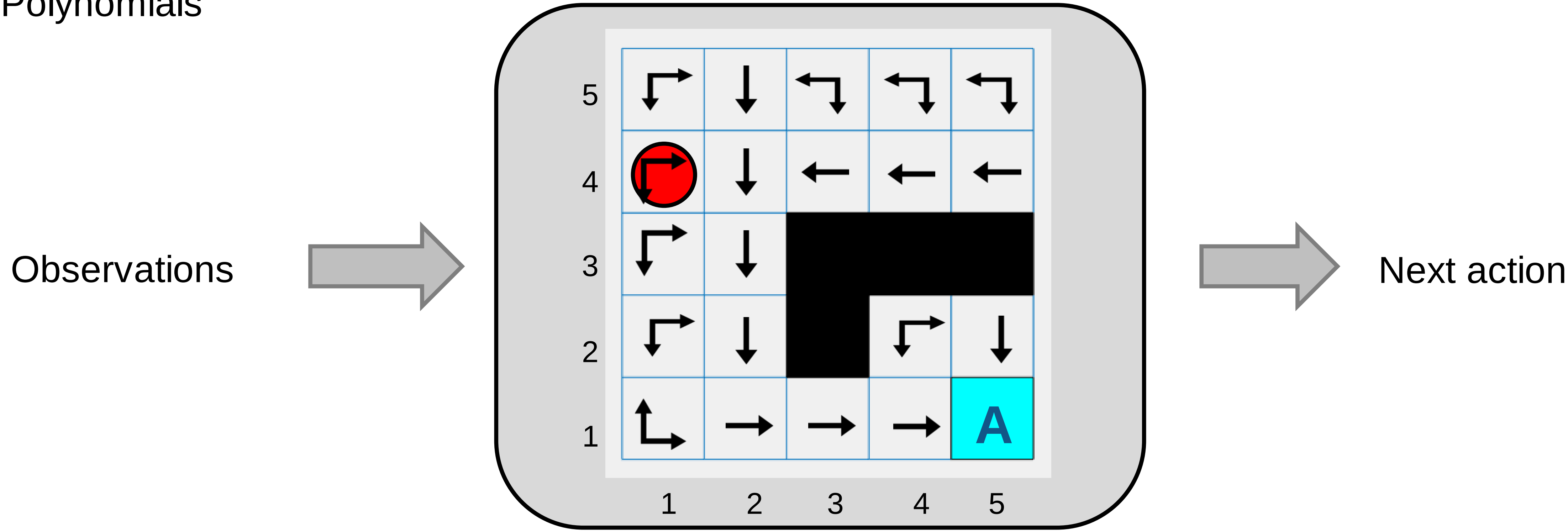
Control system	Reinforcement learning system
Adaptation mechanism	RL Algorithm
Error/Cost function	Reward
Manipulated variable	Action
Measurement	Observation
Plant	Environment
Controller	Policy

Reinforcement learning has parallels to **control system design**

Policy Representation and Deep Learning

Representation options

- Look-up table
- Polynomials

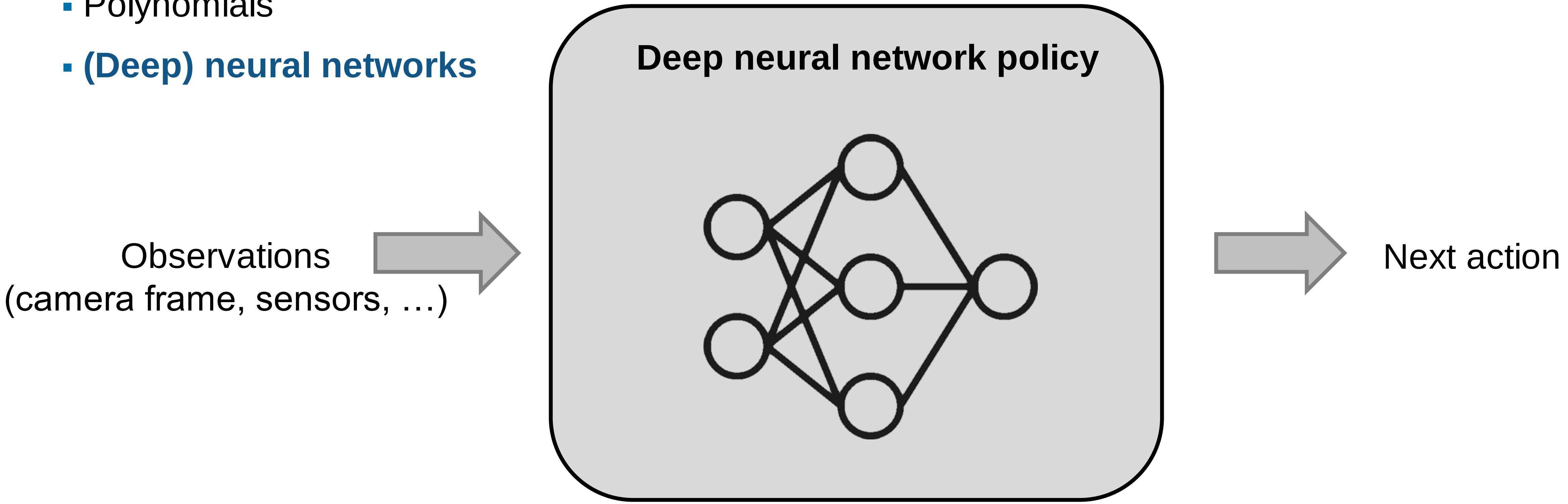


Look-up tables **do not scale** well

Policy Representation and Deep Learning

Representation options

- Look-up table
- Polynomials
- **(Deep) neural networks**



Neural networks allow representation of **complex policies**

How do I set up and solve a reinforcement learning problem?

Reinforcement Learning Workflow

- Simulation models or real hardware
- Virtual models are safer and cheaper

- Deep network? Table? Polynomial?

- Select training algorithm
- Tune hyperparameters

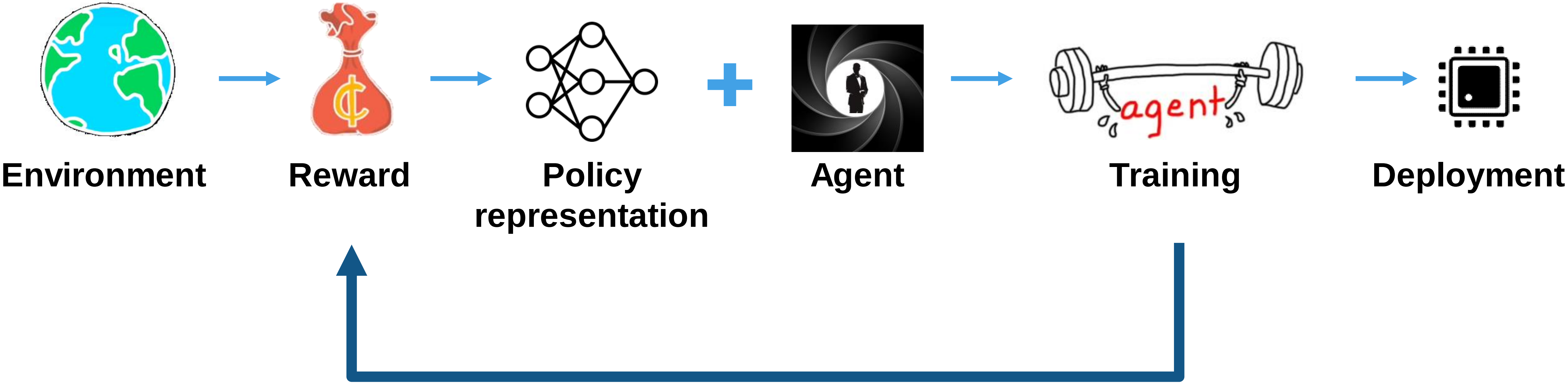
- Trained policy is a standalone function

Environment → **Reward** → **Policy representation** + **Agent** → **Training** → **Deployment**

- Numerical value that evaluates goodness of policy
- Reward shaping can be challenging

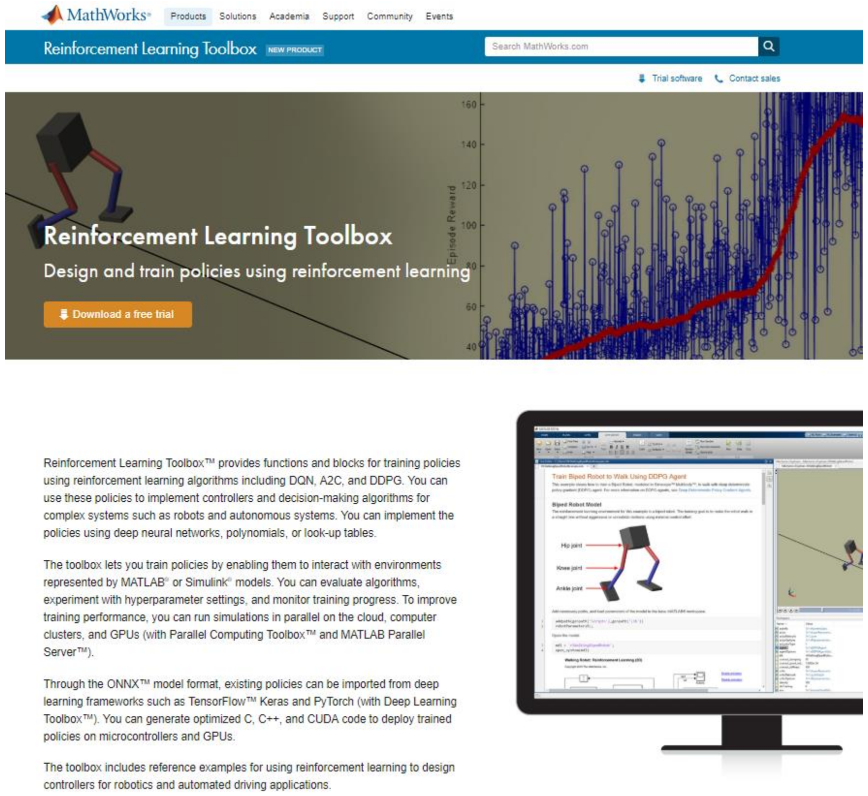
- Large number of simulations needed
- Parallel & GPU computing can speed up training
- Training could still take hours or days

Reinforcement Learning Workflow



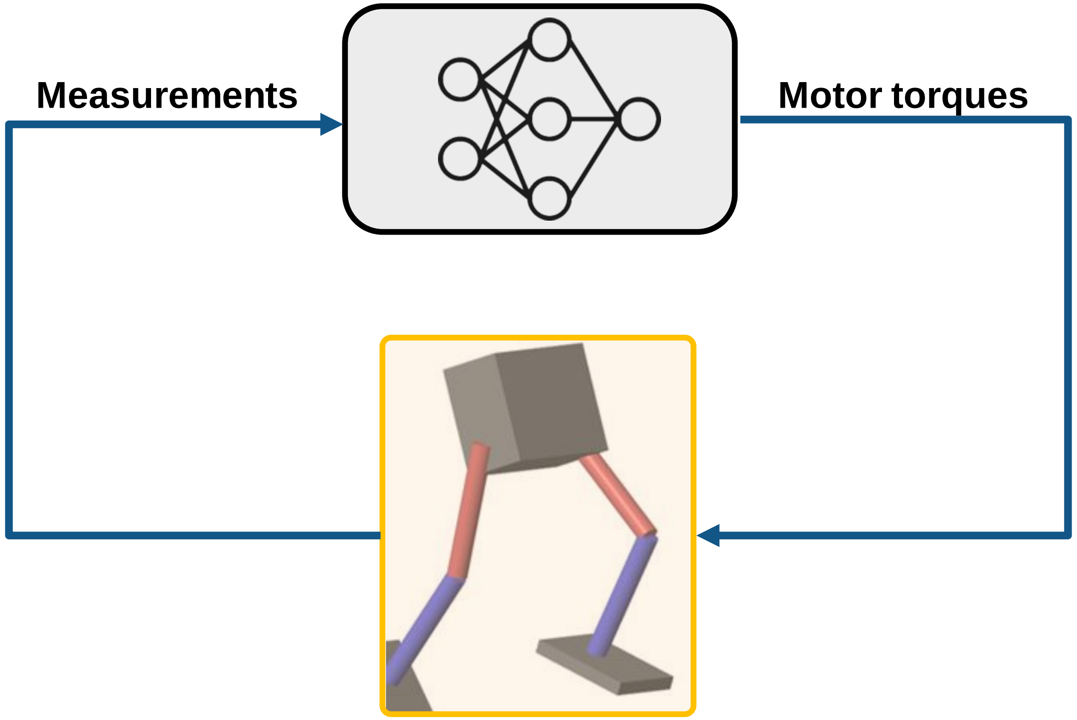
Reinforcement Learning Toolbox Introduced in R2019a

- Built-in and custom reinforcement learning algorithms
- Environment modeling in MATLAB and Simulink
  - Existing scripts and models can be reused
- Deep Learning Toolbox support for representing policies
- Training acceleration with Parallel Computing Toolbox and MATLAB Parallel Server
- Deployment of trained policies with GPU Coder and MATLAB Coder
- Reference examples for getting started



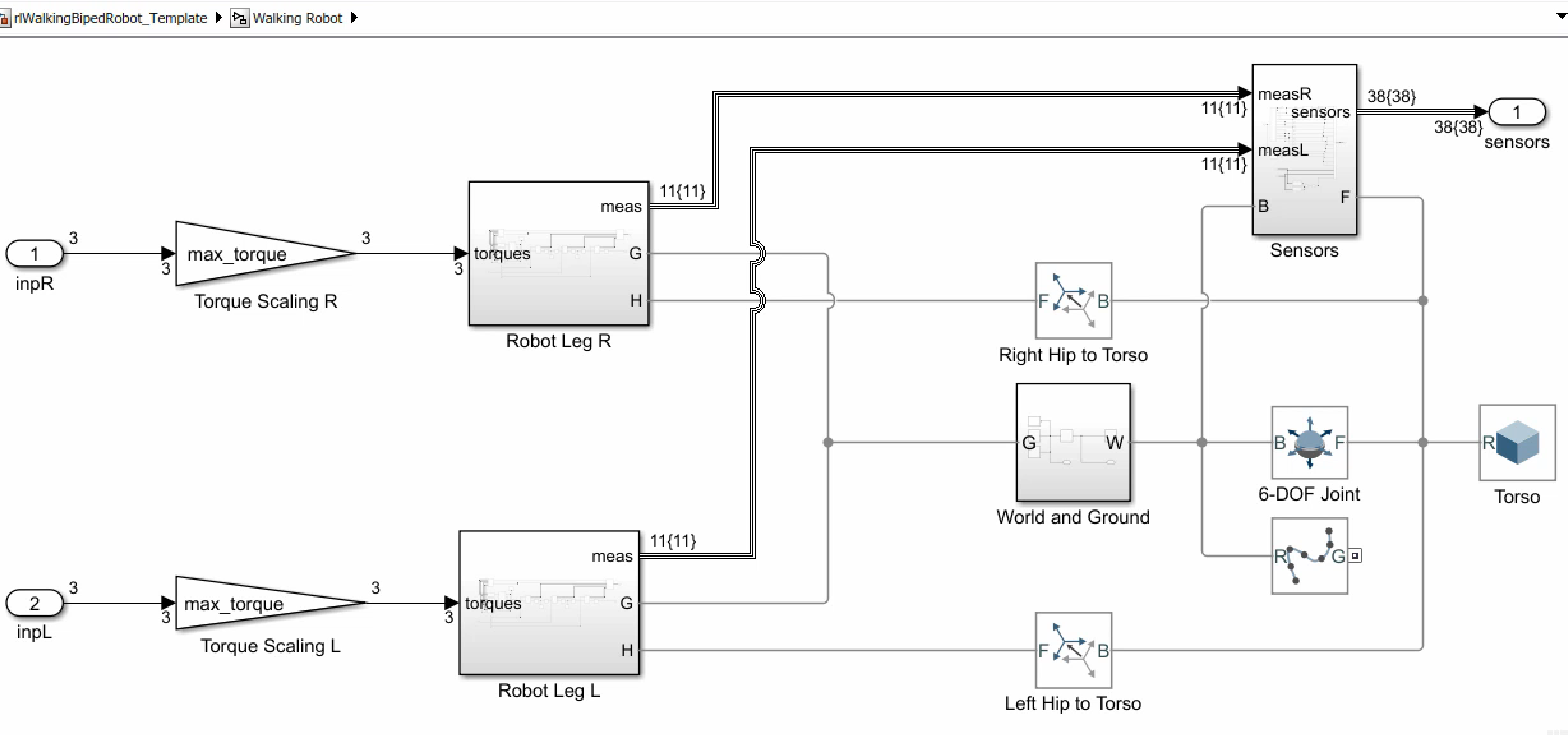
Example: Walking Robot

Environment → Reward → Policy + Agent → Training → Deployment

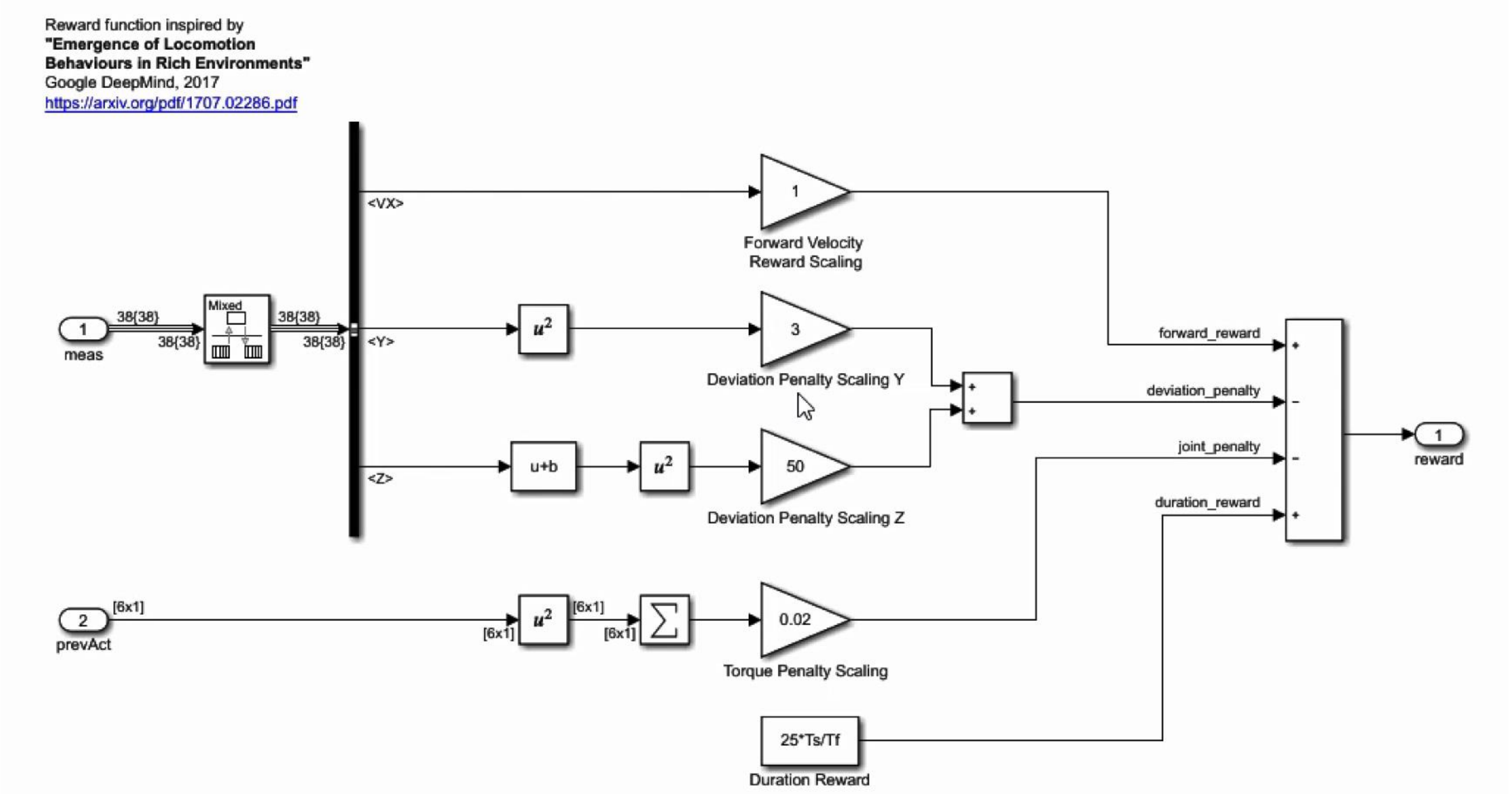


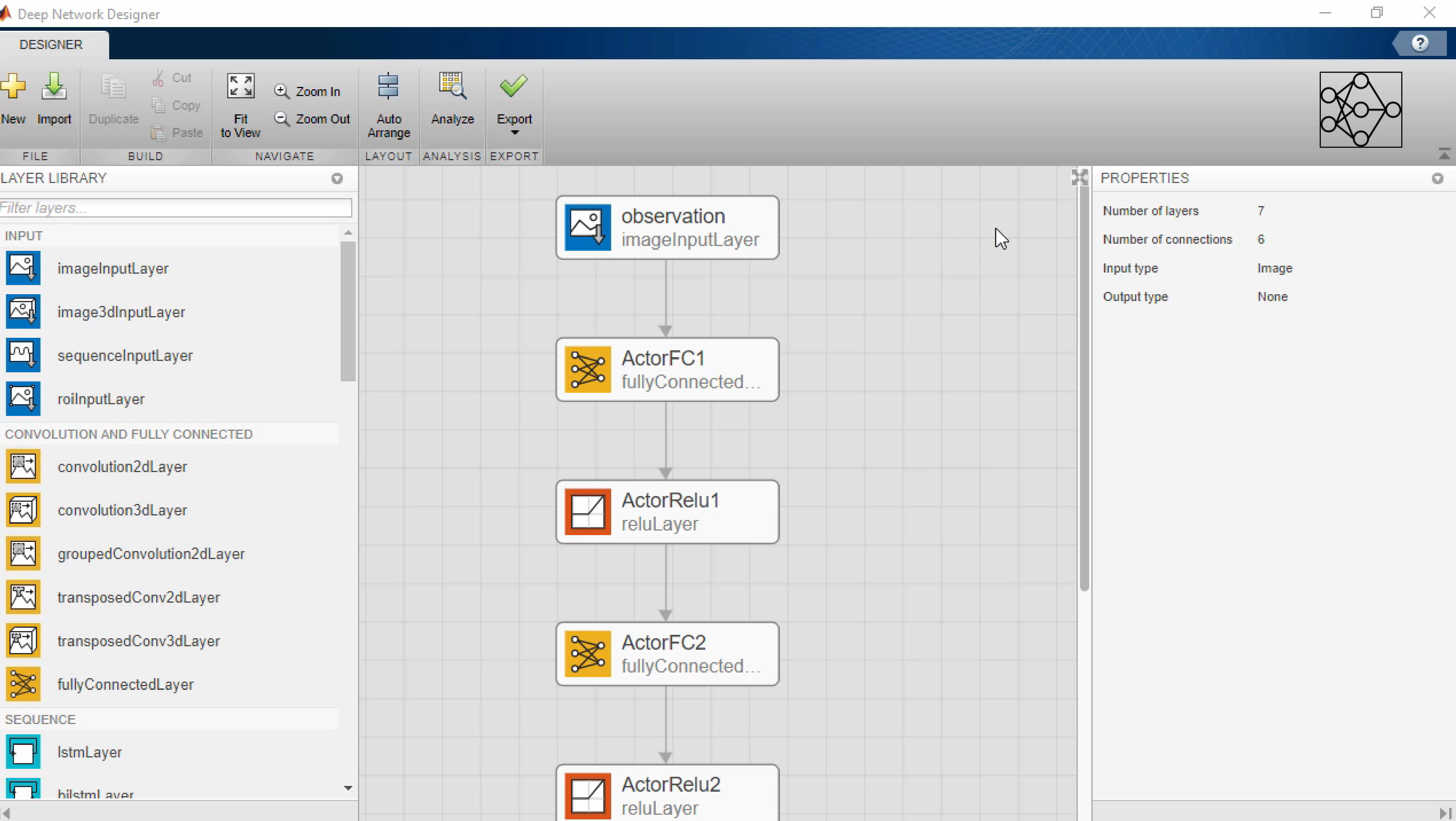
Control objective: Walk on a straight line

Creating the Environment

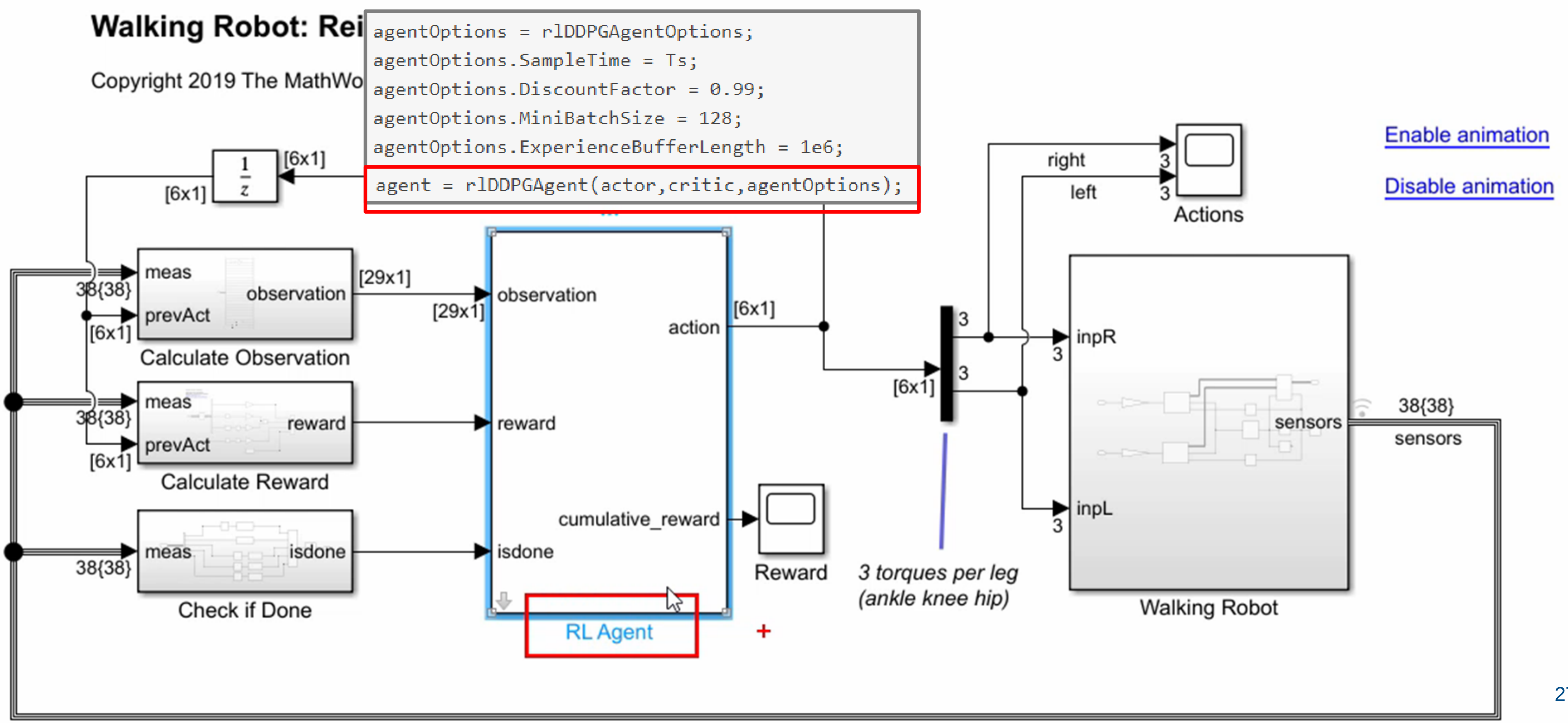


Reward Shaping




Deep Network Designer
DESIGNER
New
Import
Duplicate
Cut
Copy
Paste
Fit to View
Zoom In
Zoom Out
Auto Arrange
Analyze
Export
FILE
BUILD
NAVIGATE
LAYOUT
ANALYSIS
EXPORT
LAYER LIBRARY
Filter layers...
INPUT
imageInputLayer
image3dInputLayer
sequenceInputLayer
roiInputLayer
CONVOLUTION AND FULLY CONNECTED
convolution2dLayer
convolution3dLayer
groupedConvolution2dLayer
transposedConv2dLayer
transposedConv3dLayer
fullyConnectedLayer
SEQUENCE
lstmLayer
observation
imageInputLayer
ActorFC1
fullyConnected...
ActorRelu1
reluLayer
ActorFC2
fullyConnected...
ActorRelu2
reluLayer
PROPERTIES
Number of layers 7
Number of connections 6
Input type Image
Output type None

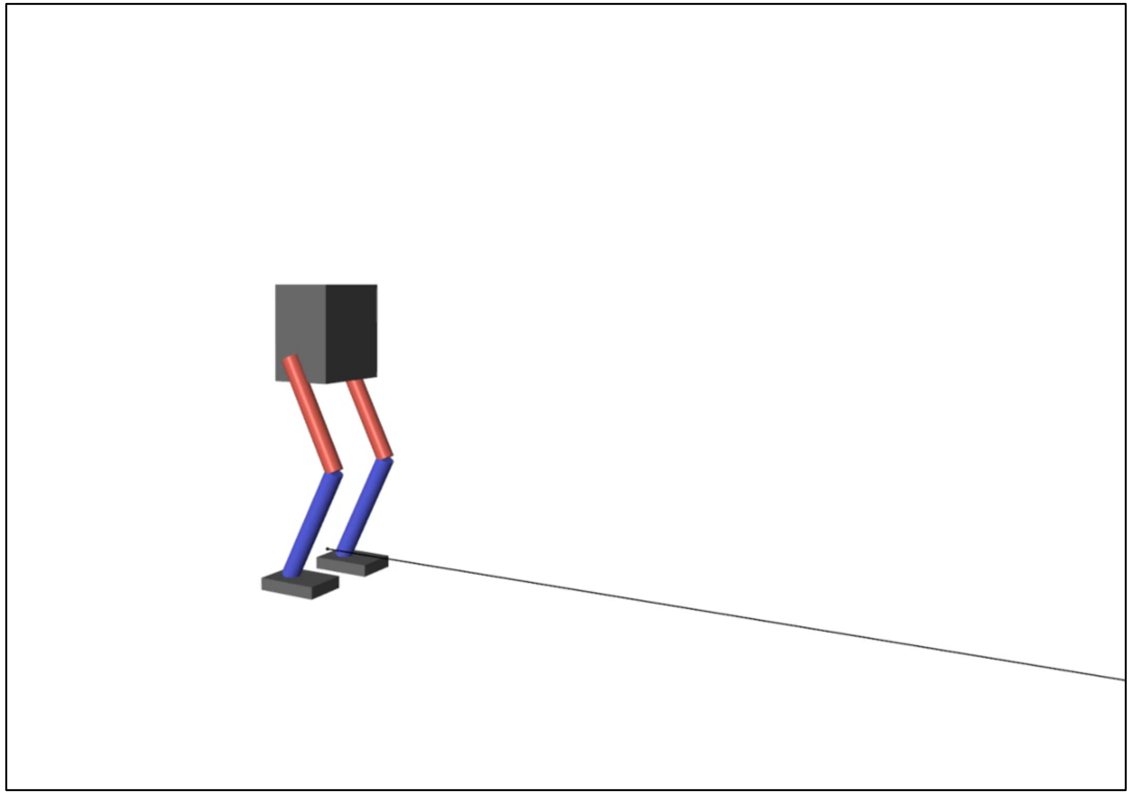
Creating the Agent

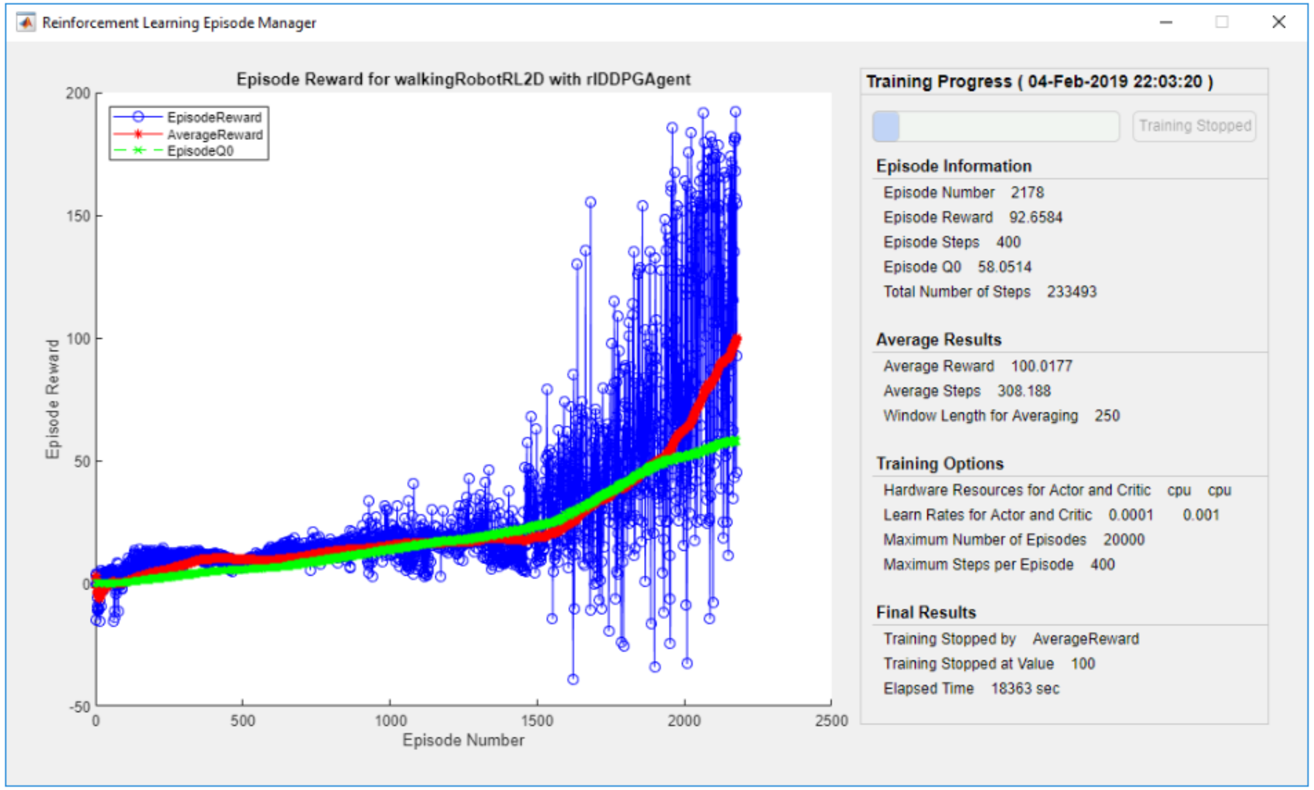


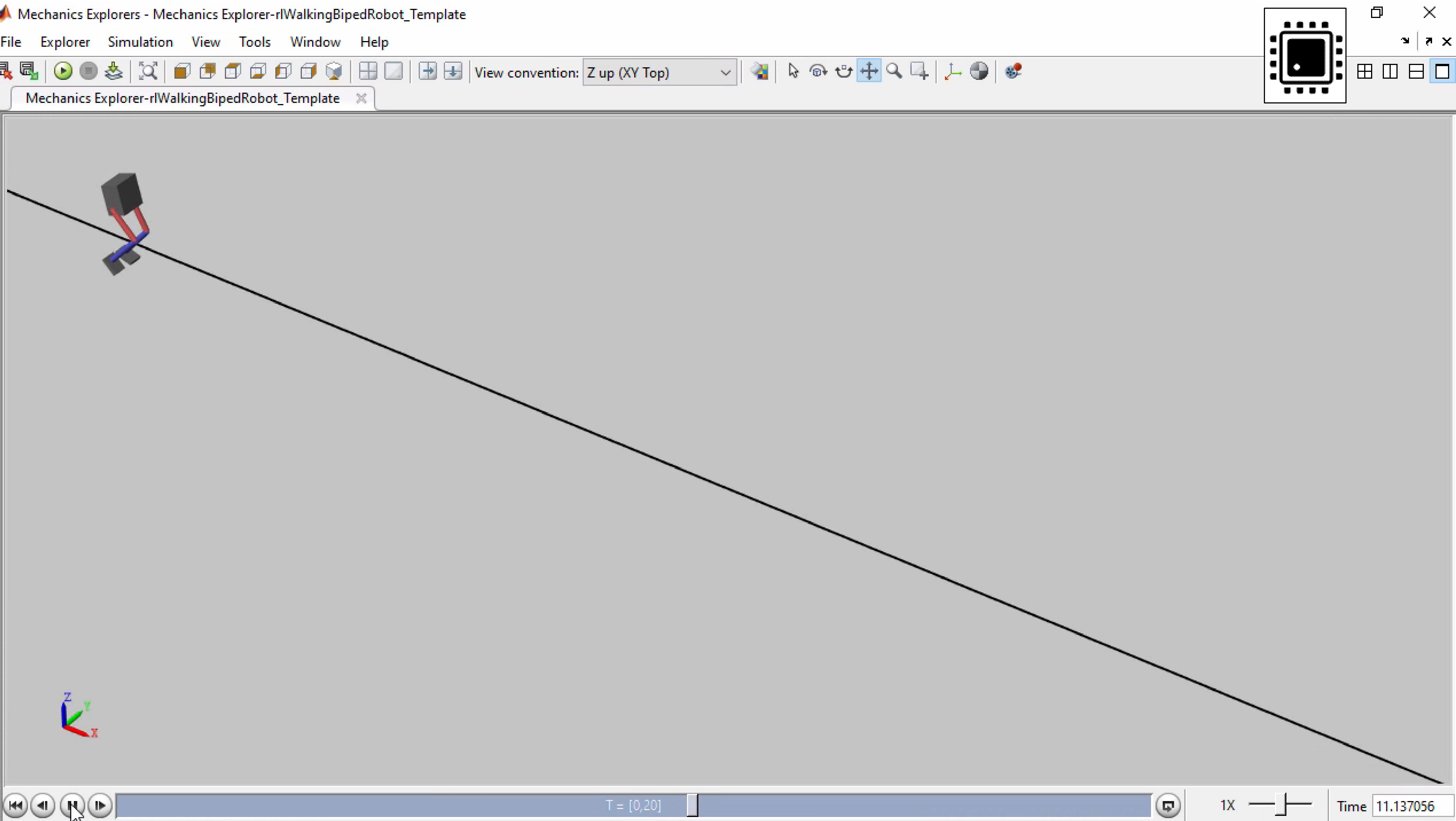

```
agentOptions = rlDDPGAgentOptions;
agentOptions.SampleTime = Ts;
agentOptions.DiscountFactor = 0.99;
agentOptions.MiniBatchSize = 128;
agentOptions.ExperienceBufferLength = 1e6;
agent = rlDDPGAgent(actor,critic,agentOptions);
```

Training the Agent

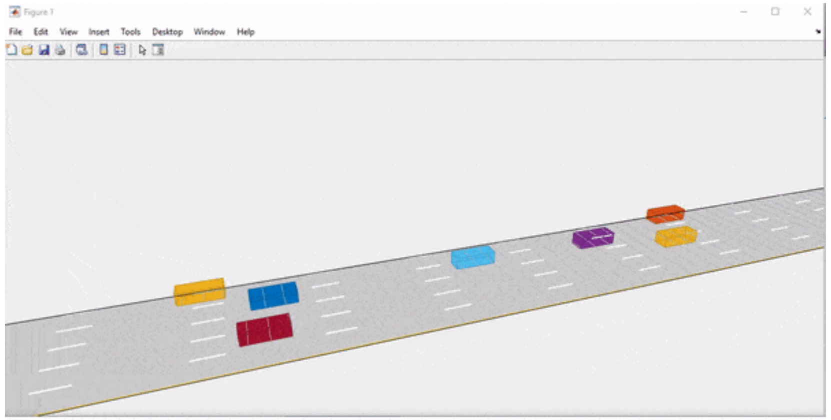

```
trainOpts.UseParallel = true;
trainOpts.ParallelizationOptions.Mode = 'async';
```

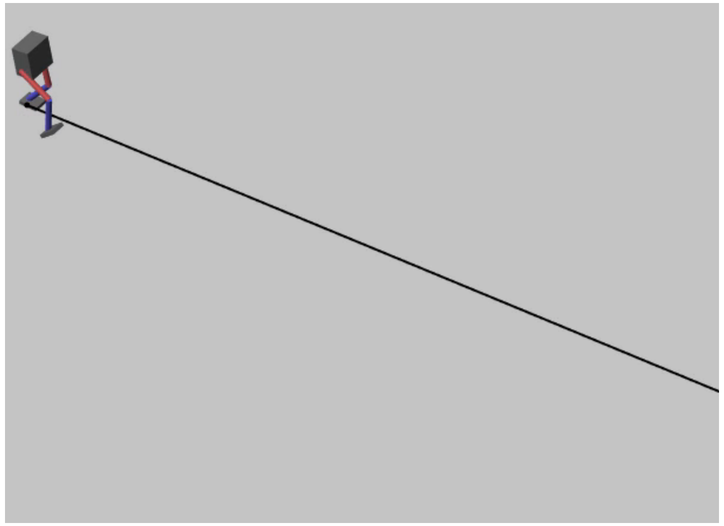


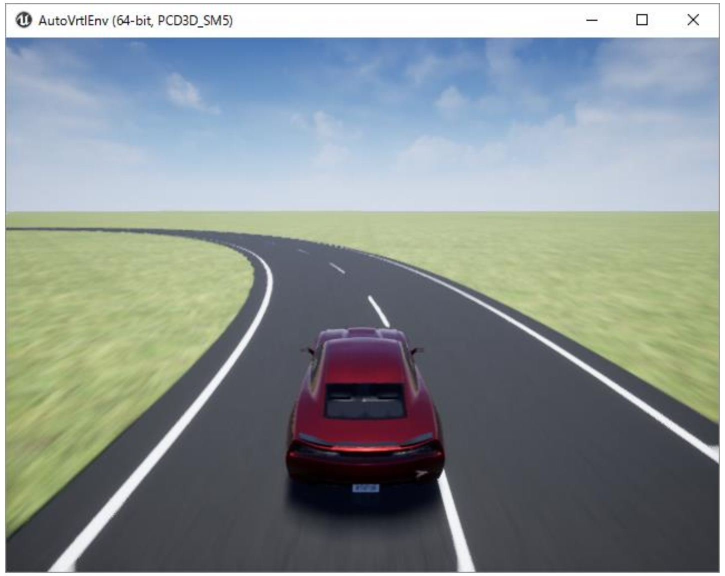


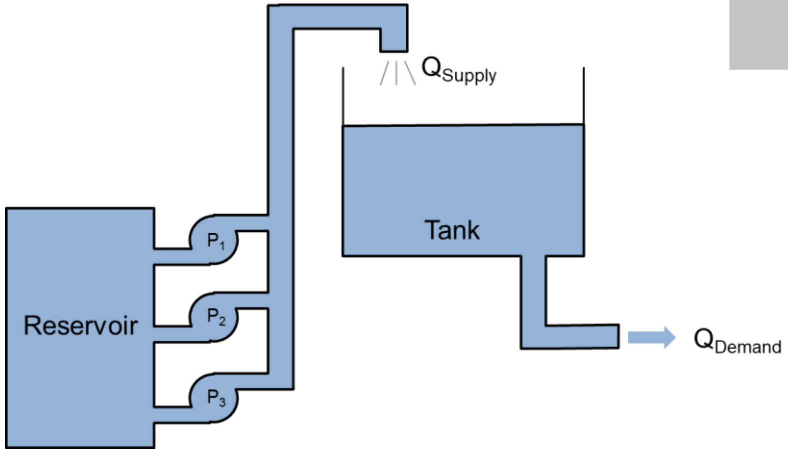

Mechanics Explorers - Mechanics Explorer-rlWalkingBipedRobot_Template
File
Explorer
Simulation
View
Tools
Window
Help
View convention:
Z up (XY Top)
Mechanics Explorer-rlWalkingBipedRobot_Template
Z
Y
X
T = [0,20]
1X
Time
11.137056

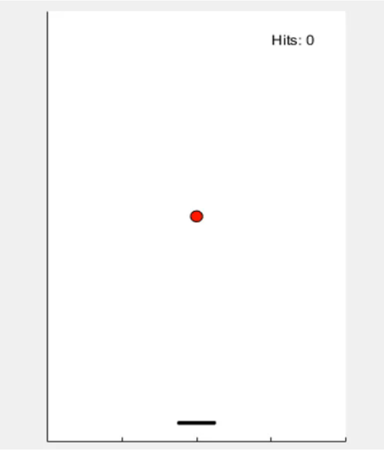
Applications of Reinforcement Learning

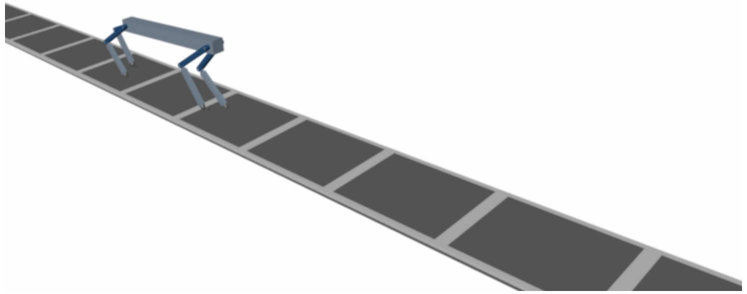




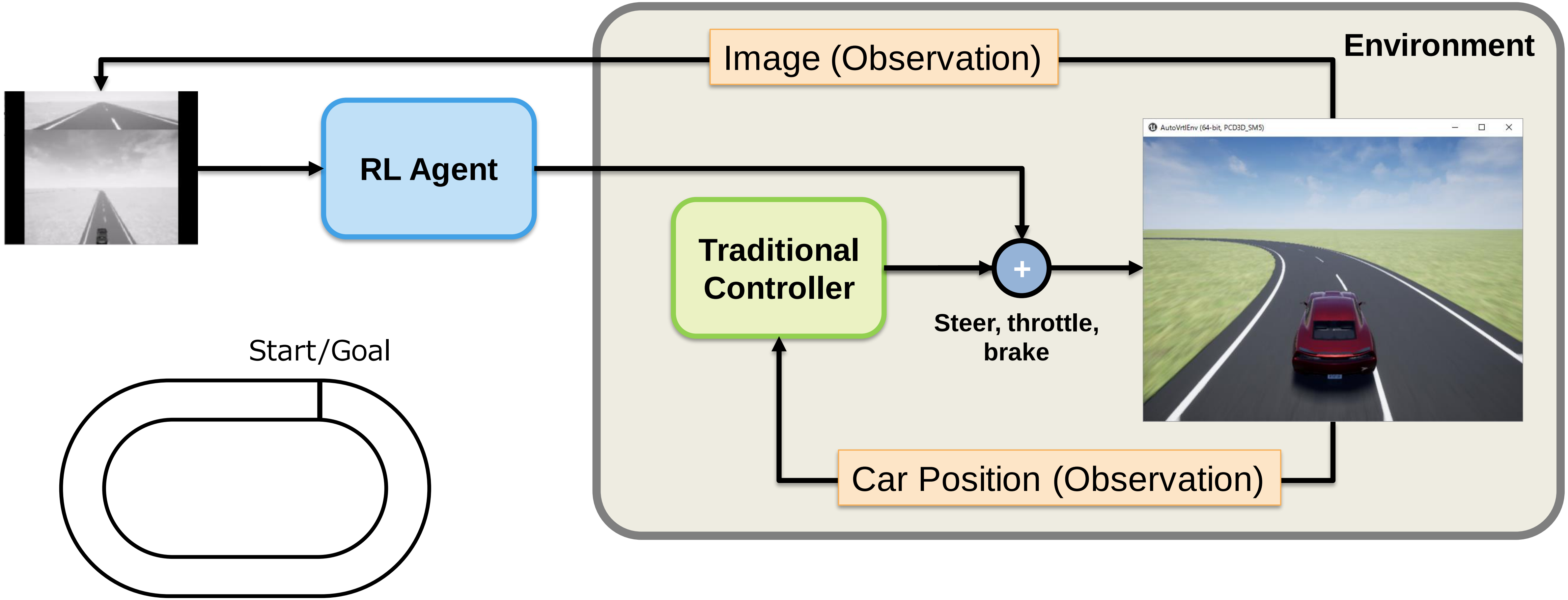




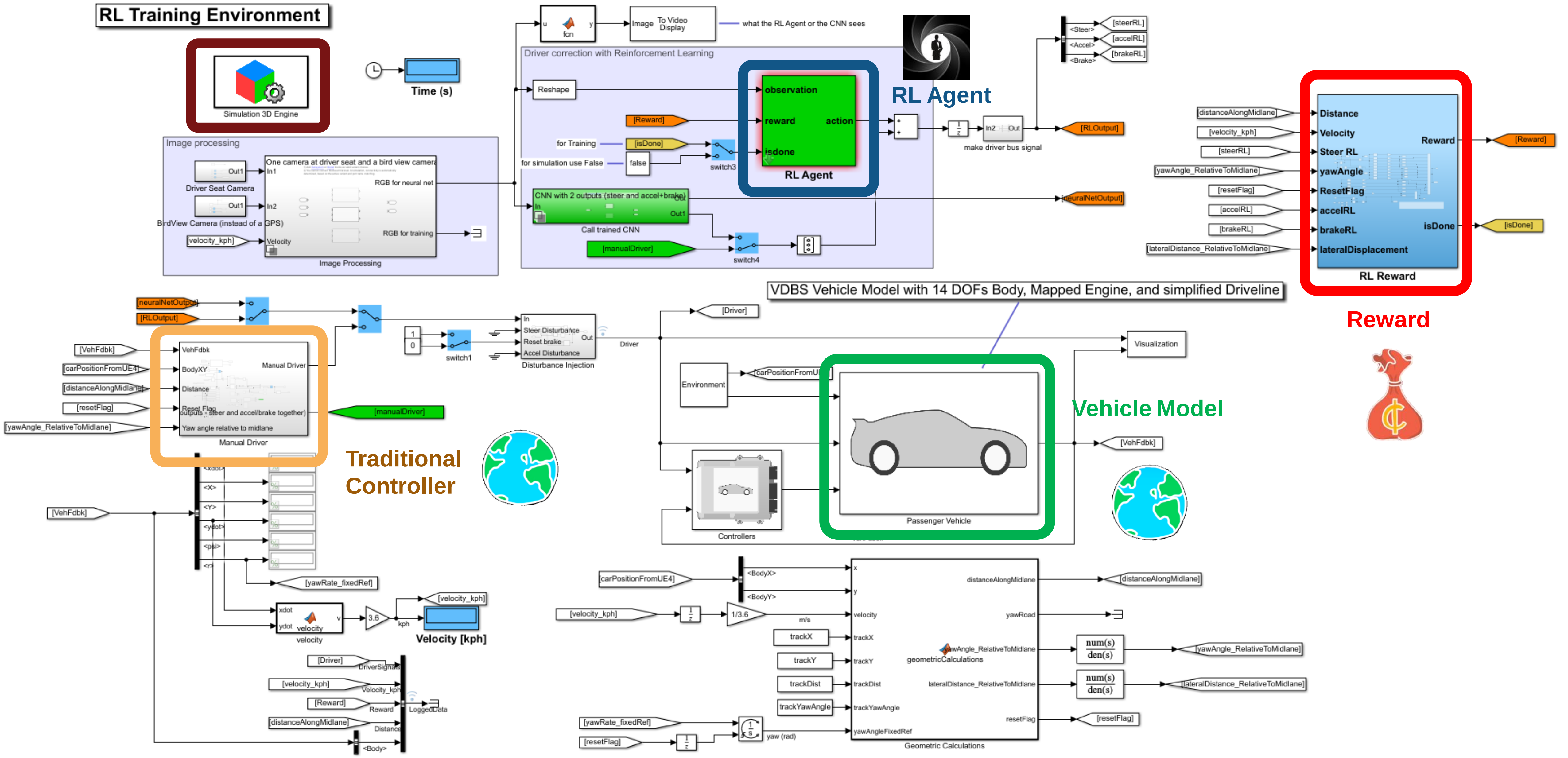




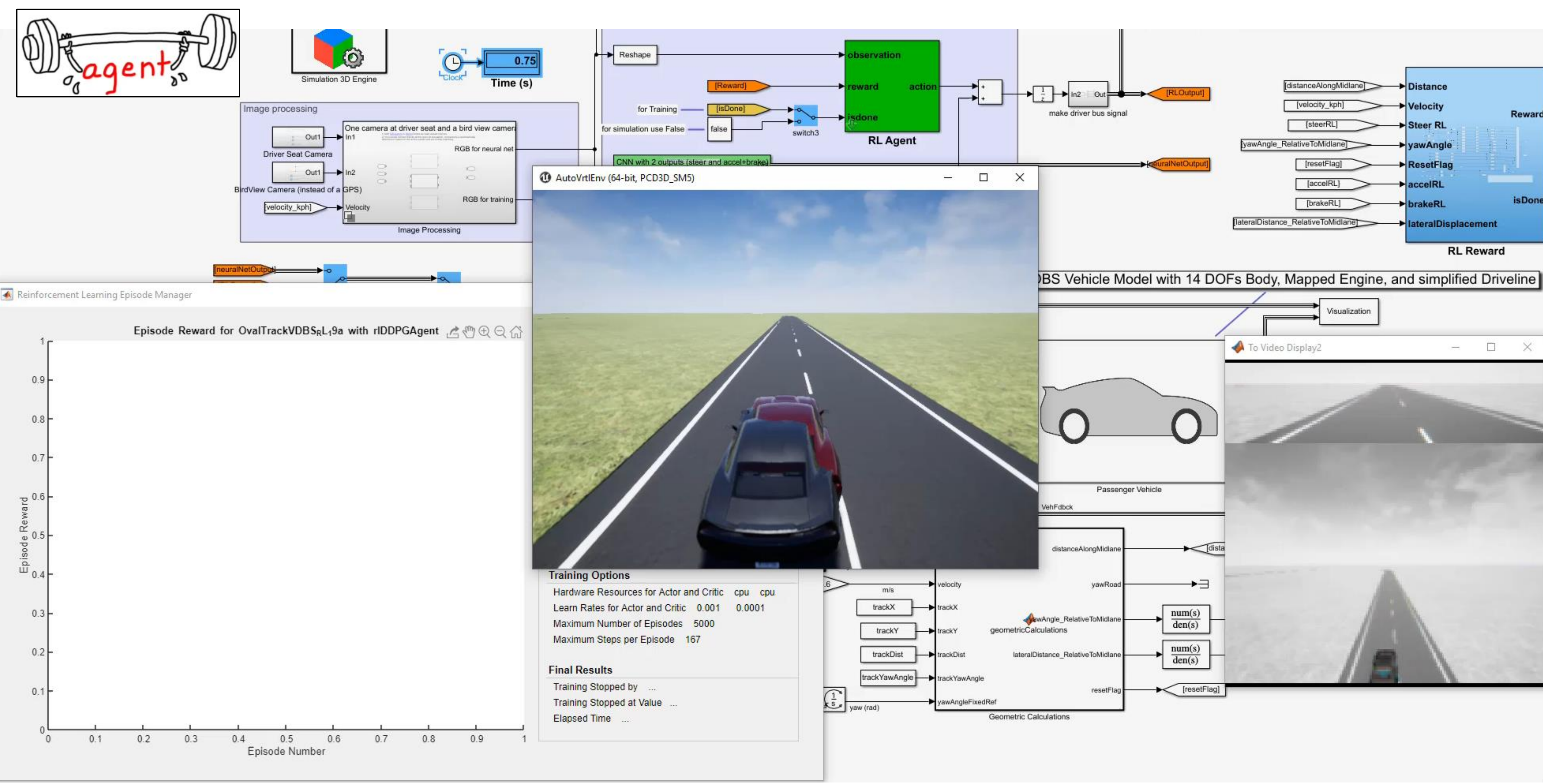
Autonomous Driving Example



Objective: Augment traditional controller with reinforcement learning to improve lap time

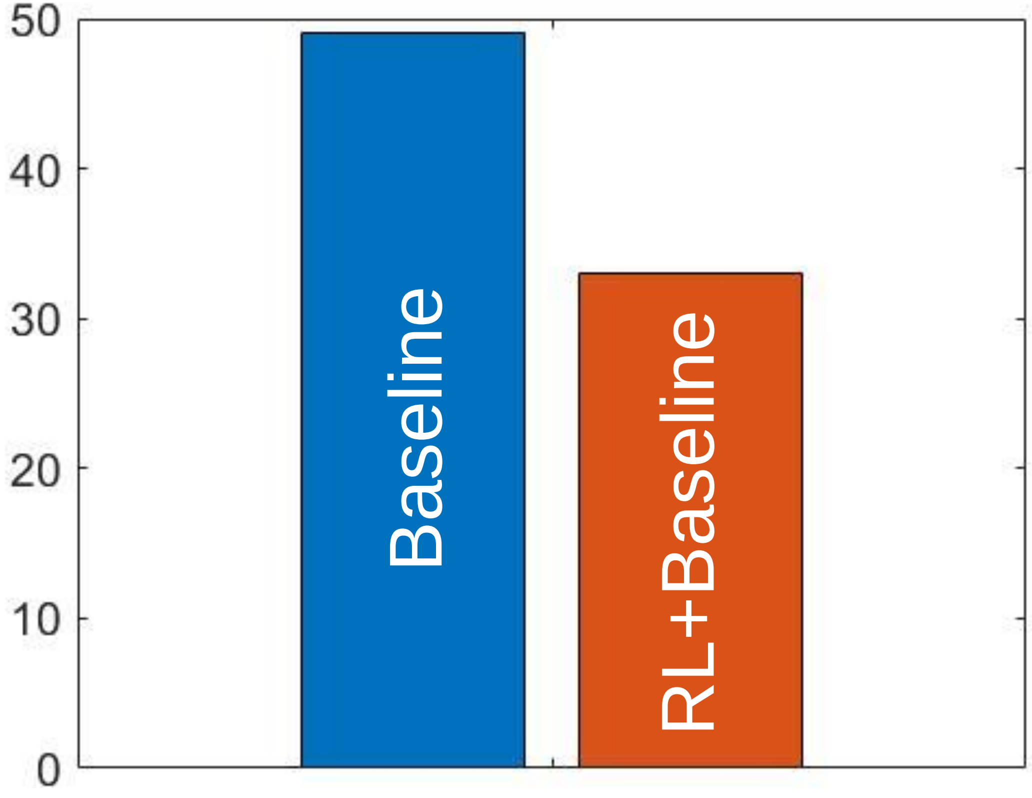


RL Training Environment
Simulation 3D Engine
Time (s)
Image processing
Driver Seat Camera
BirdView Camera (instead of a GPS)
One camera at driver seat and a bird view camera
RGB for neural net
RGB for training
Image Processing
Image To Video Display
what the RL Agent or the CNN sees
Driver correction with Reinforcement Learning
Reshape
[Reward]
for Training
[isDone]
for simulation use False
false
switch3
observation
reward
action
isdone
RL Agent
RL Agent
CNN with 2 outputs (steer and accel+brake)
Call trained CNN
[manualDriver]
switch4
make driver bus signal
[RLOutput]
[neuralNetOutput]
[steerRL]
[accelRL]
[brakeRL]
Distance
Velocity
Steer RL
yawAngle
ResetFlag
accelRL
brakeRL
lateralDisplacement
Reward
isDone
RL Reward
Reward
VDBS Vehicle Model with 14 DOFs Body, Mapped Engine, and simplified Driveline
Manual Driver
Traditional Controller
Disturbance Injection
Environment
Controllers
Passenger Vehicle
Visualization
Vehicle Model
[VehFdbk]
Velocity [kph]
Geometric Calculations
LoggedData


agent
Simulation 3D Engine
Clock
0.75
Time (s)
Image processing
Driver Seat Camera
BirdView Camera (instead of a GPS)
[velocity_kph]
One camera at driver seat and a bird view camera
RGB for neural net
RGB for training
Velocity
Image Processing
Reshape
[Reward]
for Training
[isDone]
for simulation use False
false
switch3
observation
reward
action
isdone
RL Agent
make driver bus signal
[RLOutput]
CNN with 2 outputs (steer and accel+brake)
[neuralNetOutput]
[distanceAlongMidlane]
[velocity_kph]
[steerRL]
[yawAngle_RelativeToMidlane]
[resetFlag]
[accelRL]
[brakeRL]
[lateralDistance_RelativeToMidlane]
Distance
Velocity
Steer RL
yawAngle
ResetFlag
accelRL
brakeRL
lateralDisplacement
Reward
isDone
RL Reward
AutoVrtlEnv (64-bit, PCD3D_SM5)
Reinforcement Learning Episode Manager
Episode Reward for OvalTrackVDBSRL19a with rlDDPGAgent
Episode Reward
Episode Number
Training Options
Hardware Resources for Actor and Critic cpu cpu
Learn Rates for Actor and Critic 0.001 0.0001
Maximum Number of Episodes 5000
Maximum Steps per Episode 167
Final Results
Training Stopped by ...
Training Stopped at Value ...
Elapsed Time ...
VDBS Vehicle Model with 14 DOFs Body, Mapped Engine, and simplified Driveline
Visualization
To Video Display2
Passenger Vehicle
VehFdbck
distanceAlongMidlane
yawRoad
velocity
trackX
trackY
trackDist
trackYawAngle
yawAngleFixedRef
yawAngle_RelativeToMidlane
geometricCalculations
lateralDistance_RelativeToMidlane
resetFlag
[resetFlag]
Geometric Calculations
yaw (rad)
m/s

Results

Lap time (s)

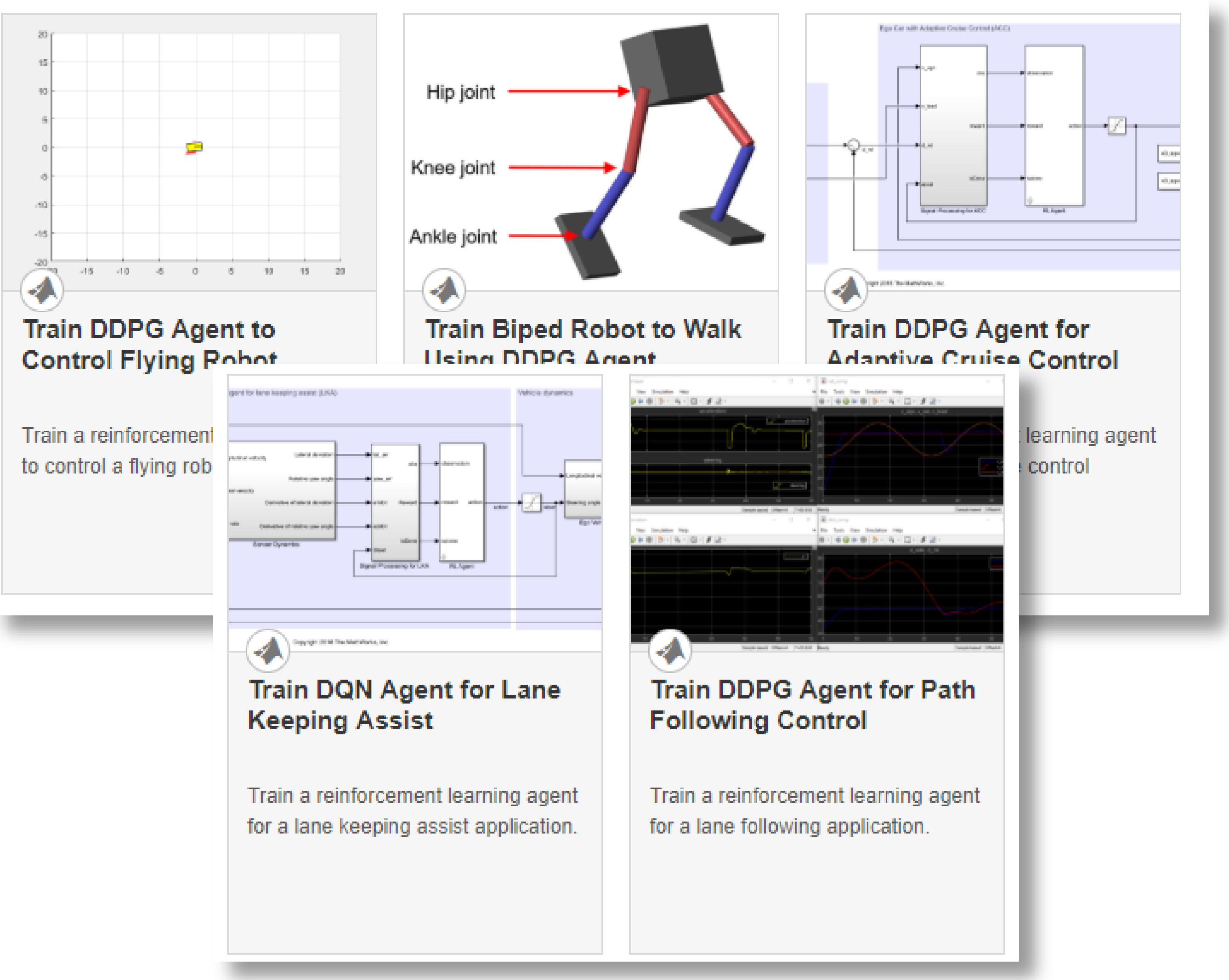


30% performance improvement

Traditional controller + reinforcement learning

Reference Applications in Documentation

- Controller Design
- Robotic Locomotion
- Lane Keep Assist
- Adaptive Cruise Control
- Imitation Learning



Pros & Cons of Reinforcement Learning

Pros

- **No data** required before training
- **New possibilities** with AI for hard-to-solve problems
- Complex **end-to-end** solutions can be developed
- **Uncertain**, **nonlinear** environments can be used

Cons

- Trained policies are **hard to verify** (no performance guarantees)
- Many trials/data points required (**sample inefficient**)
  - Training with real hardware can be expensive and dangerous
- Large number of **design parameters**
  - Reward signal
  - Network architectures
  - Training Hyperparameters

Simulations are key in Reinforcement Learning

How Can MATLAB and Simulink Help?

Challenges

- Trained policies are **hard to verify** (no performance guarantees)
- Many trials/data points required (**sample inefficient**)
  - Training with real hardware can be expensive and dangerous
- Large number of **design parameters**
  - Reward signal
  - Network architectures
  - Training Hyperparameters

MATLAB & SIMULINK

- **Reuse** existing code and models for environments
- Use simulations for **policy verification**
  - Simulate extreme scenarios
- Run simulation trials **in parallel** to accelerate training
- Consult Reinforcement Learning Toolbox **examples**
  - Iterative tuning with simulations

Key Takeaways

- What is reinforcement learning and why should I care about it?
- How do I set up and solve a reinforcement learning problem?
- What are some common challenges?

Learn More

- Reference examples for controls, robotics, and autonomous system applications
- Documentation written for engineers and domain experts
- Tech Talk video series on Reinforcement Learning concepts
- Reinforcement Learning ebooks available at mathworks.com