No. 11

Date 2019 · 10 · 28

Supervisory Meeting

Meeting Methods/Venue: Date: 2019.10.28 Time: 6:01pm

Wechat

Items for discussion:

1. The physical structure of the robot
2. Members in the Lab who I can work with
3. Time for me to go to the lab in Shanghai
4. Useful materials in the field.

Record of discussion:

1. Professor Rosendo currently has a robot with two legs that is currently under construction.
2. Since I come from an EE backgraund it would be nice if I work with it, as at the moment no one is controlling it.
3. I am asked that if it would be possible to go to Shanghai before the Chinese new year. Since it is just announced today that my last final exam is on Jan.02, I would go there around that
4. I could do some self-learning before that whatever is rebtive to the project.

Action list:

1. Finish the visiting student application form and submit it to the Shanghaitech University.
2. Find some learning material online.

No. 12

Date 2019 · 11 · 4

Experiment: Simulation of Walking Robot Based on MATLAB

I try to do some exercises according to the tutorials "MATLAB and Simulink Robotics Arena" (https://www.youtube.com/playlist?list=PLn8PRpmsu08oLuta YWEvcue8Rq7q407D"). First, I begin with a walking robot model based on Simulink, given by the tutorial. I explore the model, run the simulation and try to understand each part of the model, as shown in the Figures followed. In addition, plots of angles of each joint (ankle roll, ankle pitch, knee, hip pitch, hip roll, and hip yaw) for right and left legs are also attached. (Total time costs: around 3 hours, including the set up of the env.)

8 pm

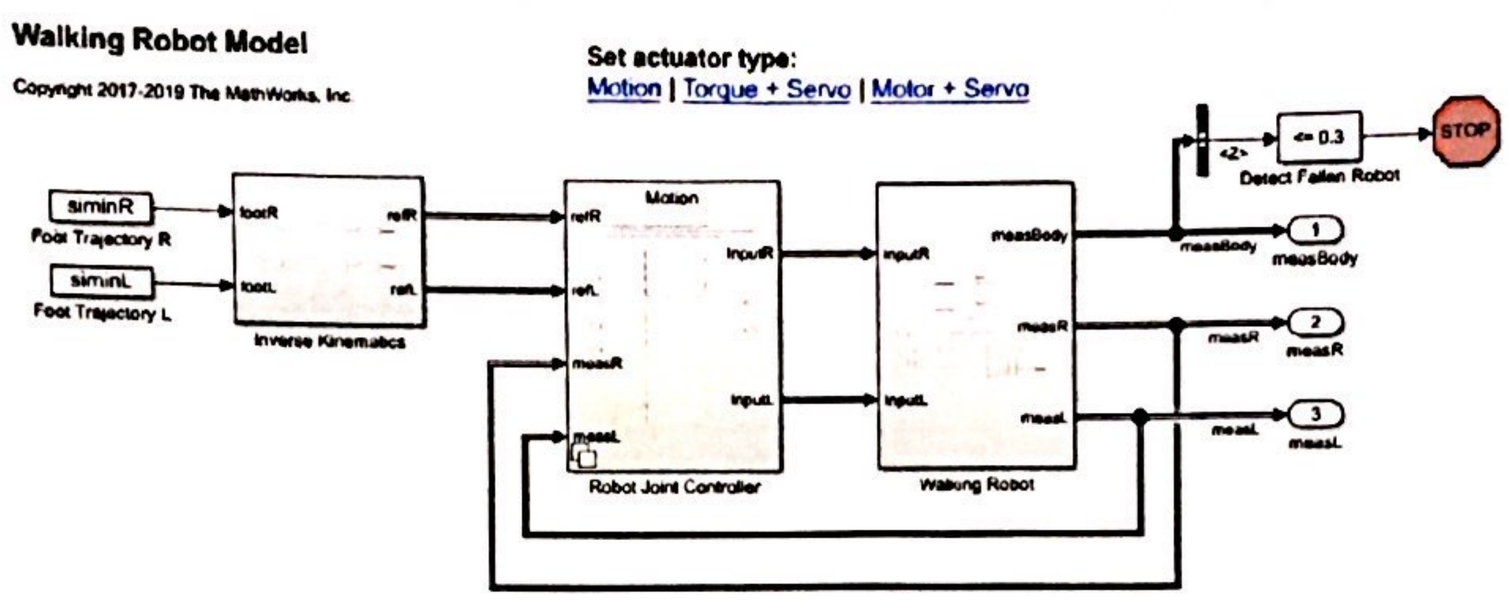


Figure 1. Result of training a walking robot with DDPG on Python (average reward per episode).

Figure 2 Simulation result of the model.

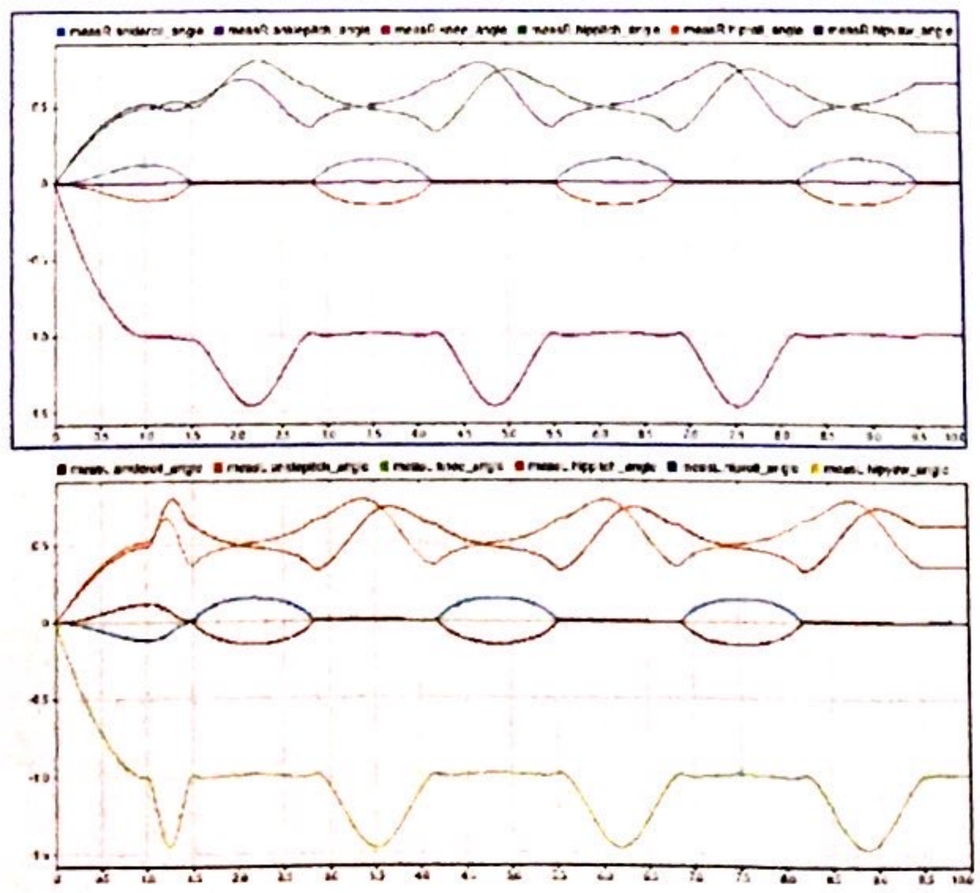


Figure 3. Angles for right leg (upper) and left leg (lower).

Notes on Introduction to Reinforcement Learning

This is the note for the first lecture of the cource Introduction to Reinforcemeny Learning online by David Silver from UCL.
(Https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ.)
ps: The cources cover most of the basic concepts of RL, which is necessary and meaningful to reflesh again and again while diving into more advanced topic.

- History: the sequence of observation, actions, rewards $H_t = A_1, O_1, R_1, \cdots, A_t, O_t, R_t$
- State: the information used to determine what happens next, $S_t = f(H_t)$ (summary of history)
  - Environment state
  - Agent state: the agent's internal representation

- A state S_t is Markov if and only if $P(S_{t+1} | S_t) = P(S_{t+1} | S_1, \cdots, S_t)$

- Full observability: Agent state = environment state = information state ⇒ Markov decision process
- Partial observability: Agent indirectly observes environment ⇒ Partially observable Markov decision process
  - Complete history: $S_t^a = H_t$
  - Beliefs of environment state: $S_t^a = (P[S_t^e = s^1], \cdots, P[S_t^e = s^n])$
  - Recurrent neural network: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

Policy is the agent's behaviour
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = P[A = a | S = s]$

Value function is a prediction of future reward

Model predicts what the environment will do next
- Transitions: predicts the next state $P_{ss'}^a = P[s' = s' | S = s, A = a]$
- Rewards: predicts the next reward $R_s^a = E[R | S = s, A = a]$

* Prediction: evaluate the policy
* Control: find the best policy

No. 14

Date 2019 · 11 · 20

Experiment: Traning a Walking Robot on MATLAB (GA)

Equipment: PC (Require MATLAB 2019b or later)

Files: createSmoothTrajectory.m,

doSpeedupTasks.m

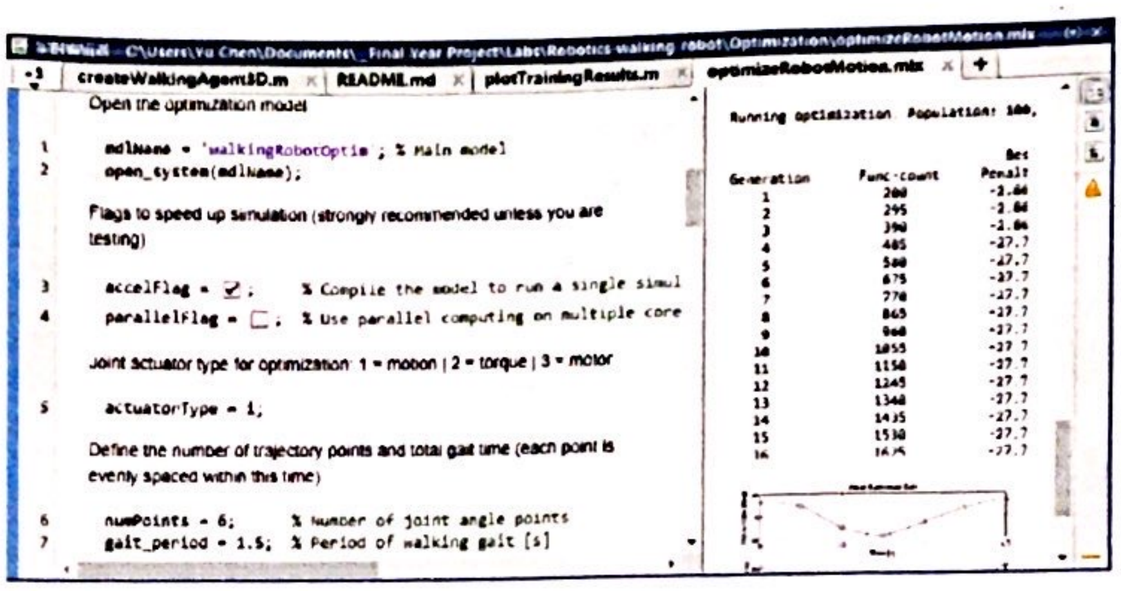
optimizeRobotMotion.mlx main optimization script

simulateWalkingRobot.m

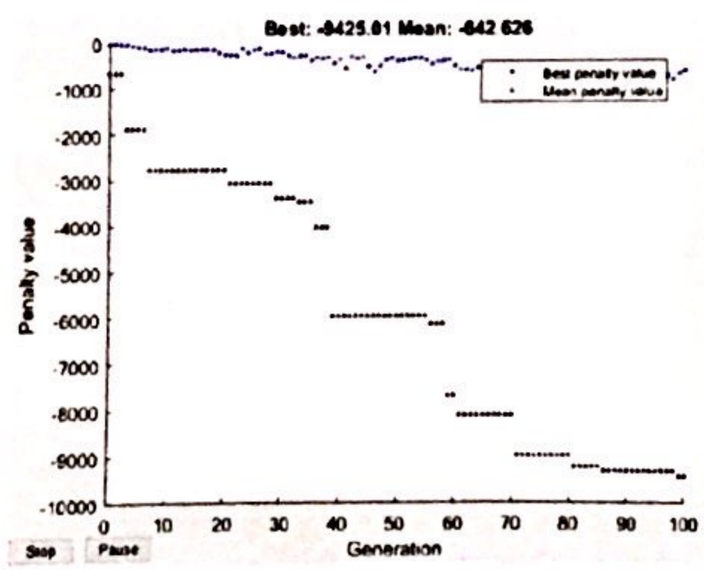
walkingRobotOptim.slx

Process:

① Tunning the parameter (6:34pm) and run to train.



② plot the learning results (6:58pm)



No. 15

Date 2019 · 11 · 22

Notes on Markov Decision Process

This is the notes for the second lecture of the cource Introduction to Reinforcement Learning by David Silver.

(Almost all RL problems can be formalised as MDPs)

Markov Processes: State transition probability: $P_{ss'} = \mathbb{P}(S_{t+1}=s' \mid S_t=s)$ $\Rightarrow$ transition matrix P

A Markov Process (Markov Chain) is a tuple $\langle \vec{S}, \hat{P} \rangle$

Markov Reward Process $\Rightarrow$ a Markov chain with values $\Rightarrow$ a tuple $\langle S, P, R, \gamma \rangle$

(Immediate) Reward $R_s = \mathbb{E}(R_{t+1} \mid S_t=s)$

Return $G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ (in one sample)

Value Function $v(s) = \mathbb{E}(G_t \mid S_t=s)$

Bellman Equation for MRPs $v(s) = \mathbb{E}(R_{t+1} + \gamma v(S_{t+1}) \mid S_t=s) = R_s + \gamma \sum P_{ss'} v(s')$

Markov Decision Process $\Rightarrow$ a Markov Reward Process with decision $\Rightarrow$ a tuple $\langle S, A, P, R, \gamma \rangle$

Policy $\pi(a \mid s) = \mathbb{P}(A_t=a \mid S_t=s)$ - A policy fully defines the behaviour of an agent

- MDP plices depend on the current state (not the history)

MDP $\rightarrow$ MRP $S_1, R_2, S_2, R_2 \cdots \langle S, P^{\pi}, R^{\pi}, \gamma \rangle \Leftarrow$ $\begin{cases} P^{\pi}_{ss'} = \sum_{a \in A} \pi(a \mid s) P^{a}_{s,s} \\ R^{\pi}_{s} = \sum_{a \in A} \pi(a \mid s) R_s^a \end{cases}$

$\begin{cases}$ State-value function $v_{\pi}(s) = \mathbb{E}_{\pi}(G_t \mid S_t=s)$ $\rightarrow$ how good is it in state s when following policy π

Action-value function $q_{\pi}(s,a) = \mathbb{E}_{\pi}(G_t \mid S_t=s, A_t=a)$ $\rightarrow$ Starting from state s, taking action a, and then following policy π

Bellman Expectation Equations $\begin{cases} v_{\pi}(s) = \mathbb{E}_{\pi}(R_{t+1} + \gamma V_{\pi}(S_{t+1}) \mid S_t=s) = \sum_{a \in A} \pi(a \mid s)(R_s^a + \gamma \sum_{s' \in S} P^a_{ss'} V_{\pi}(s')) \\ q_{\pi}(s,a) = \mathbb{E}_{\pi}(R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a) = R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} \sum_{a' \in A} \pi(a' \mid s') q_{\pi}(s', a') \end{cases}$

Optimal Value Function $v_*(s) = \max_{\pi} v_{\pi}(s)$ $\quad q_*(s,a) = \max_{\pi} q_{\pi}(s,a)$

Optimal Policy $\pi \geq \pi'$ if $v_{\pi}(s) \geq v_{\pi'}(s), \forall s$

There is always a deterministic optimal policy for any MDP $\pi_*(a \mid s) = \begin{cases} 1 & \text{if } a = \underset{a \in A}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$

Bellman Optimality Equation $\begin{cases} v_* = \max_a q_*(s,a) \\ q_*(s,a) = R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} v_*(s') \end{cases} \Rightarrow v_*(s) = \max_a R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} v_*(s')$

No 16

Date 2019 · 11 · 28

Notes on Dynamic Programing

This is the notes for the third lecture of the cource Introduction to Reinforcement Learning by David Silver.

problems with properties { Optimal substructure
Overlapping subproblems ⇐ Bellman equation gives recursive decompoition

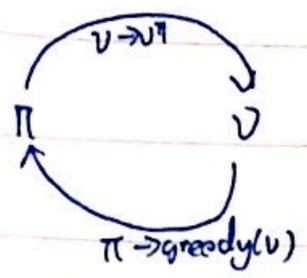
{ Prediction MDP $\langle S,A,P,R,\gamma\rangle$ and pdicy π → value function v_π
Control MDP $\langle S,A,P,R,\gamma\rangle$ → optimal value function v_*, optimal policy π_*

Pdicy Evaluation : Evaluate a given pdicy π $v_1 \to v_2 \to \cdots \to v_\pi$

Synchronous backups : each interation, all states $v_{k+1}(s) = \sum_{a\in A} \pi(a|s)(R_s^a + \gamma \sum_{s'\in S} P_{ss'}^a v_k(s'))$ $\vec{v}^{k+1} = \vec{R}^\pi + \gamma \vec{P}^\pi \vec{v}^k$

Policy Iteration Given a policy π → Evaluate the policy $v_\pi(s) = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s]$ ←

Improve the policy by action greedily with respect to v_π $\pi' = greedy(v_\pi)$



$\pi'(s) = \underset{a\in A}{\operatorname{argmax}}\, q_\pi(s,a)$

Improvement step → $v_\pi(s) = \max_{a\in A} q_\pi(s,a)$ the Bellman optimality equation has been satisfied

Value Interation Principle of Optimality: A policy $\pi(a|s)$ achieves the optimal value from state s, $v_\pi(s) = v_*(s)$

it and only if { for any state s' reachable from s
π achieves the optimal value from state s', $v_\pi(s') = v_*(s')$

⇐: if we know the solution to subproblems $v_*(s')$ —one-step lookahead→ $v_*(s) \leftarrow \max_{a\in A} R_s^a + \gamma \sum_{s'\in S} P_{ss'}^a \cdot v_*(s')$

Interative application of Bellman optimality backup → find optimal policy π

$v_1 \to v_2 \to \cdots \to v_*$ update $v_{k+1}(s)$ from $v_k(s')$

(http://www.cs.ubc.ca/~poole/demos/mdp/vi.html)

Summary Prediction: Interative Policy Evaluation (Bellman Expectation Equation)

Control { Policy Interation (Bellman Expectation + Greedy Policy Improvement)
Value Interation (Bellman Optimality Equation)

Algorithms based on { state-value function $v_\pi(s)$ or $v_*(s)$: Complexity $O(mn^2)$ per interation, m actions n states
action-value function $q_\pi(s,a)$ or $q_*(s,a)$: Complexity $O(m^2n^2)$ per interation

Organisation of Literatures

This note is for the Interim Report.

"Percepton and navigation in unknown environments: the DAPPA robotics challenge" E.J.Molinos [Molinos2013]

→ The paper presents different techniques to achieve the tasks proposed in DAPPA.

(Not very useful because of the application limitation.)

"A small humanoid robot sdr-4x for entertainment applications" M.Fujita

→ It describes a new small humanoid type robot, SDR-4X, for entertainment purpose.

(It could be used as a reference when dealing with similar robots.)

"Bipedal Robots: Modeling, Design and Walking Synthesis" C.Chevallereau

→ A book published in 2007 introducing bipedal robot in terms of modeling, design and walking synthesis

(A very comprehensive introduction, but just limits state-of-the-art methods)

"Zero-moment point thirty-five years of its life" M.Vukobratovic

→ It is devoted to the permanence of the concept of Zero-Moment Point.

(It is a good material to learn ZMP)

"Modifiable walking pattern of a humanoid robot by using allowable ZMP variation" B.J.Lee

→ It proposes a novel algorithm that can modify a walking period and a step length in both sagittal and lateral planes.

(An application of ZMP)

"Strategies for adjuting the ZMP reference trajectory for maintaining balance in humanoid walking" K.Nishiwaki

→ It addresses strategies of changing the reference trajectories of the future ZMP that are used for online repetitive walking pattern generation.

"The development of honda humanoid robot" K.Hirai 1998

→ It presents the mechanism, system configuration, basic control algorithm and integrated function of the Honda humanoid robot.

(a really classic robot design)

"Optimizing foot centers of pressure through force distribution in a humanoid robot" P.M.Wensing 2013

→ It presents two formulation of the FDP for humoids in double suppor, and propose objective functions within a general frame-work to address the variety of competing requirements for the realization of balance.

"Multi-contact bipedal robotic locomotion" H.Zhao 2019

→ It presents a formal framework for achieving multi-contact bipedal robotic walking.

(Traditional method to control AMBER2 and ATRIAS)

"Toward an expressive bipedal robot: variable gait synthesis and validation in a planar model" U.Muzaita2018

→ It presents a framework for stylistic gait generation in a compass-like under-actuated planar biped model.

(Stylish control)

No. 18

Date 2019 · 12 · 3

" Real time motion generation and control for biped robot -4th report : integrated balance control"

→ It propoes a design technique for feedback gains to stabilize the upper body potion under varying vertical ground reaction force.

" Robust feedback control of ZMP-based gait for the humanoid robot Nao

→ It proposes an approach combines the ZMP stability criterion with angular momentum suppresion and step timing contrd.

C ! it points out the importance of working in physical platform !)

" Towards natural bipedal walking : virtual gravity compensation and capture point control "

→ It proposes a use of virtual gravity compensation (VGC), essentially a dynamic controller that outputs joint torques, to generate the force and momere as desired.

" Motion and walking stablization of humanoids using sensory reflex control," J. Kim 2016

→ It employs and elaborates on sensory reflex contrd to stabilize standing motion and biped walking using basic sensors such as an inertial measurement unit (IMU) and a force-sensing resistor (FSR)

" Observer-based postural balance control for humanoid robots " M. Stachbazi 2013

→ It presents an observer-based balancing control strategy that is robust to posistent perturbations on the ground slope which is based on the estimation of the disturbance without the need for an inertial measurement unit.

[6] " * " Machine learning algorithms in bipedal robot control " S. Wang 2012

→ A 2012-published paper reviewing the state-of-the-art learning algorithms and their applications to bipedal robot control.

(A really useful survey and the structure of the paper is inspiring !)

* " Combining model-based policy search with online model learning for control of physical humanoids." I. Mordatch 2016

→ It presents an automatic method for interactive control of physical humanoid robots based on high-level tasks.

(Intrac An really intreasting and inspiring method that trained on both simulated and real robots in different phases.

" Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance: Experiments on the standing posture "

→ It uses CTRNN and BPTT for online learning control ~~tasks~~.

ANN, balance

" Biologically inspired control system for 3D locomotion of a humanoid biped robot " A.A. Saputra 2016

→ It proposes a control system for 3-D locomotion of a humanoid biped robot based on a biological approach.

RNN, stabilization system

" Rejection of an external force in the sagittal plane applied on a biped robot using a neuro-fuzzy controller " J.P. Ferreira 2007

→ It describes the control of an autonomous biped robot capable to be subjected to external forces applied in the sagittal plane.

(Cortious)

"A new hybid intelligent contrd algorithm for a seven-link biped walking robot" F. Faradpour 2014

→ It proposes the implementation of a new hybirid intelligent contrd approach for a seven-link biped walking robot.
(A method between classic contrd method and learning method.

* "Gait balance and acceleration of a biped robot based on Q-Learning" K.S. Hwang 2016

→ It presents a method for the biped dynamic walking and balance contrd using reinforcement learning.
(Train on both simulation and reality environment!!)

"Motion segmentation and balancing for a biped robot's imitation learning" K.S. Hwang 2017

→ It propaes an imitating algorithm called posture-based imitation with badance learning (Post-BL).
(It works in physical robot)

"Posture self-stabilizer of a bipedal robot based on training platform and reinforcement learning" W. Wu 2017

→ An automatic abstraction method for state space is propaed by using the Gauss basis function and inner evaluation indexs to speed up the learning process. (only in simulation)

* "Survey of model-based reinforcement learning: application on robotics" A.S. Polydoros 2017

→ A comprehensive survey of model-based RL that applied on robotics
(It's a really nice summary of RL algorithm, which could be used as a index of RL literature search. [4,17,59,61,63]) →Walking-

"PILCO: a model-based and data-efficient approach to policy search" M.P. Deisenroth 2011

→ Introduction of a new RL algorithm
(A 2011 paper that first introduces this method.)

"Efficient reinforcement learning for robots using informative simulated priors" M. Cutler 2015

→ It presents a novel method method for transfering data from a simulator to a robot, using simulated data as a prior for real-world learning. (Improved PILCO)

"Model-based imitation learning by probabilistic trajectory matching" P. Englert 2013

→ It presents an imitation-learning approach to efficiently learning a task from expert demonstrates.
(An application of PILCO)

"Improving PILCO with Bayesian Neural Network Dynamics Models" Y. Gal 2017

→ It extends PILCO's framework to use Bayesian deep dynamics models with approximate variational inference, allowing PILCO to scale linearly with number of trials and observation space dimensionality*

"Integral control of humanoid balance" B. Stephens 2007

→ It presents a balance controller that allows a humanoid to recover from large disturbances and still maintain an upright posture.

No. 20

Date 2019 · 12 · 4

* "Reinforcement learning in robotics: a survay" J. Kober 2013

→ It attempts to strengthen the links between the two reserch communities (robotics, RL) by providing a survey of work in reinforcement learning for behaviour generation in robots.

"Multi-task policy search for robotics" [Deisenroth 2014]

→ An application of PLICO

(It contains a brief and useful introduction of RL)

"Learning to contol a low-cost manipulator using data-efficient reinforcement learning" [Durrant-Whyte 2012]

→ Another application case of PLICO

(A more detail description of the last application in fact)

"Toward fast policy search for learning legged locomotion" [Deisenroth 2012]

→ It presents a data-efficient approach to learning feedback controllers for legged locomotive systems.

(The outcome is really close to the objective of my project!!)

"Model-based imitation learning by probabilistic trajectory matching" [Englert 2013]

→ It presents an imitation-learning approach to efficiently learn a task from expert demonstrations.

(It introduces an a new concept "imitation learning" but the application case is similar as before)

"Policy gadient based reinforcement learning for real autonomous underwater cable tracking"

→ It proposes a field application of a high-level RL control system for solving the action selection problem of an autonomous robot in cable tracking task.

"Comparison of model-free and model-based methods for time optimal hit control of a badminton robot

→ A model-based method is compared to model-free reinforcement learning methods.

(The developed controllers are illustrated on a simulation model of robot.)

"Survey of behavior learning applications in robotics - state of the art and perspectives."

→ It analyzes the current state of machine learning for robotic behaviours.

Not only focus on algorithms but also real world application. (a higher level of view)

"A survey of dynamic robot legged locomotion"

→ It proposes a survey focuses on legged dynamic locomotion of robots.

Main part of the paper is focus on history and introduction.

#

Draft for literature organisation

- Walking
  - six legs
    - Maes and Brooks (1990)
    - Kirchner (1997)
  - quadrupedal
    - Bicchell and Livingston (2007)
    - Kohl and Stone (2004)
    - Bartsch et al. (2016)
  - biped
    - Benbrahim and Franklin
    - Matsubara et al. (2005)
    - Geng et al. (2006)
    - Kormushev at el. (2011c)
    - Misawa and Behnke (2015)

No. 22

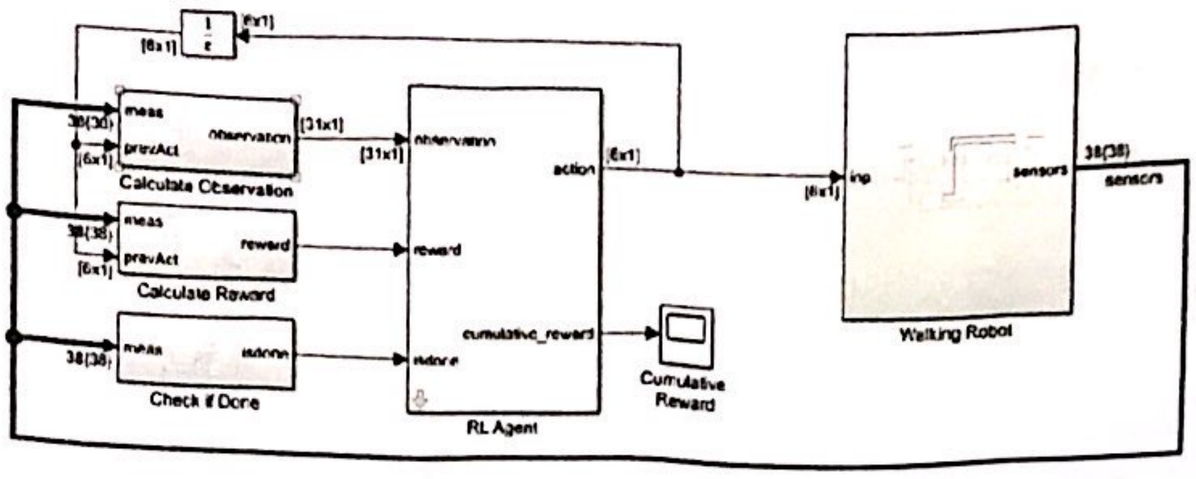
Date 2019. 12 . 10

Experiment: Training a robot on MATLAB using DDPG

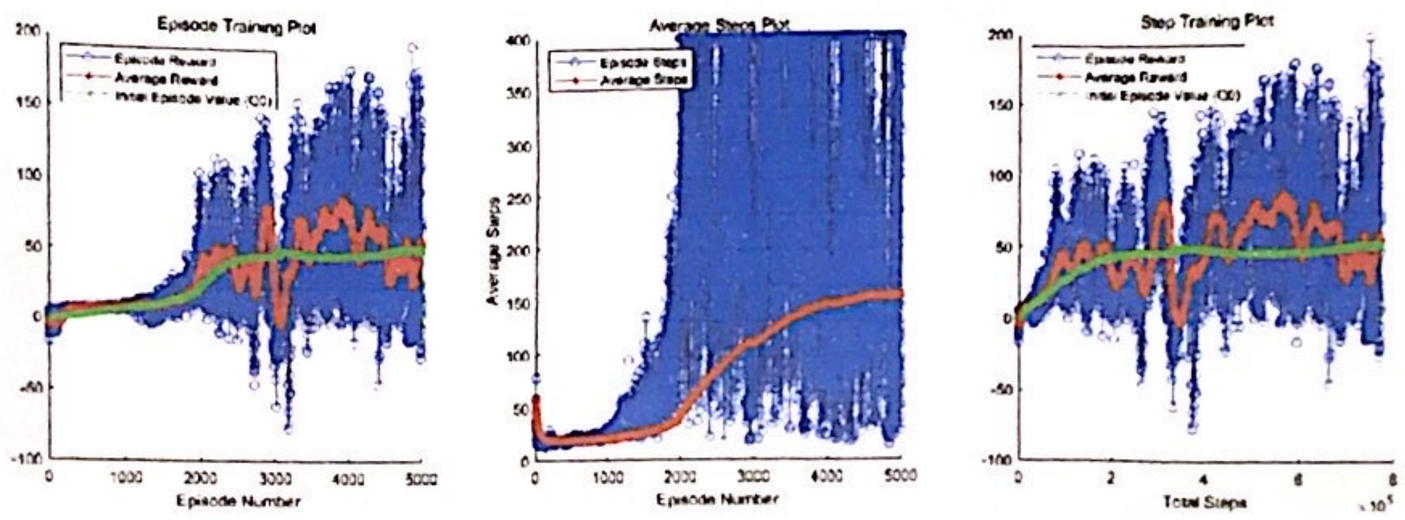
Equipment: PC (Require MATLAB R2019b or later)

Files: create DDPGNetwork.m, create DDPG Option.m, Create Walking Agent 2D.m, creat Walking Agent 3D.m, plot Training Results.m robotParameters RL.m, walkerInvKin.m, walkerResetFcn.m, walkingRobot RL2D.slx, (walking Robot RL2D.slxc), walking - RobotRL3D.slx.

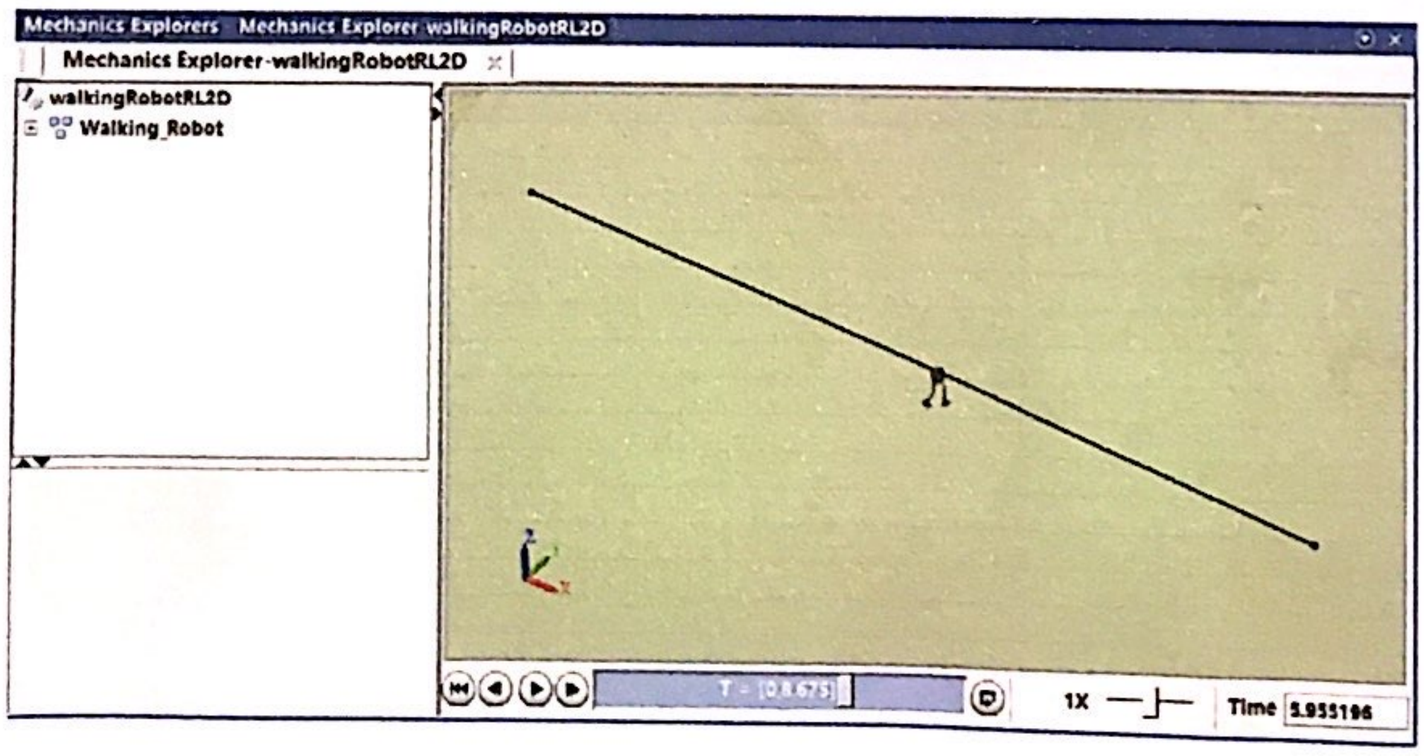
Process: ① Explode the model and run. (1:36pm)



② Got the trained results and plot it. (5:52 pm)



③ Demonstrate the simulated results. (6:12 pm)



No. 23

Date 2019 · 12 · 14

Notes on Model-Free Prediction

This is a lecture note for the fourth lecture of the cource Introduction to Reinforcement Learning by David Silver.

Model-free: no one tell the environment (Estimate the value function of an unknown MDP)

Monte-Carlo Learning (learn directly from episodes of experience)

value = mean return ⇒ can only apply MC to episodic MDPs

Goal: learn v_π from episodes of experience under policy π $S_1, A_1, R_2, \cdots, S_k \sim \pi$

Recall { return: $G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$

value function $v_\pi(s) = \mathbb{E}_\pi(G_t \mid S_t = s)$ expected return $\xrightarrow{MC}$ empirical mean

To evaluate state s: { The first time-step t that state s is visited in an episode (First-Visit MC Policy Evaluation)

Every time step t that state s is visited in an episode (Every-Visit MC Policy Evaluation)

- Increament counter $N(s) \leftarrow N(s) + 1$
- Increament total return $S(s) \leftarrow S(s) + G_t$
- Value is estimated by mean return $V(s) = S(s)/N(s)$
- By law of large numbers, $V(s) \to v_\pi(s)$ as $N(s) \to \infty$

To evaluate state s: Every time step t that state s is visited in an episode

For each state S_t with return G_t: $N(S_t) \leftarrow N(S_t) + 1$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t))$$

Temporal-Difference Learning TD learns from incomplete episodes, by bootstrapping (update a guess toward a guess)

Goal: learn v_π online from experience under policy π

{ Incremental every-visit MC: $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$

TD(0): $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$

{ MC converges to solution with minimum mean-squared error * MC does not exploit Markov property

TD(0) converges to solution of max likihood Markov model * TD exploits Markov property