Control of robotic arm with spiking neural networks

Projektpraktikum Computational Neuro Engineering 2017SS

Meng Wang, Bo Huang, Zhiwei Han

Image: References (left to right):
1) https://pixabay.com/de/sicherheit-wort-laser-modern-574079/ 2) https://pixabay.com/de/sicherheit-sicherheitskonzept-augen-1163108/
3) https://pixabay.com/de/objektiv-linse-null-eins-bin%C3%A4r-1278493/ 4) https://pixabay.com/de/sicherheit-wort-laser-modern-574079/

Content

Content

Introduction and Motivation

What

- Point its weapons (i.e. robot arm) only towards hostile intruders
- Welcome allies with a friendly greeting
- Emulation pre-calculated distance
- Implement the procedure using spiking neural network based reinforcement learning algorithm (potentially Q-Learning)
- **Overall goal: finding a policy that allows the robot to reach a specific region through massive self learning**

Why

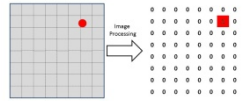
- Detection and tracking are important problems in autonomous systems
- Using spiking neural network based reinforcement learning approach we can achieve more complex or dynamic behavior

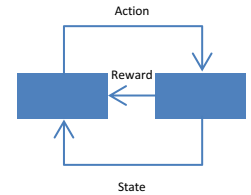
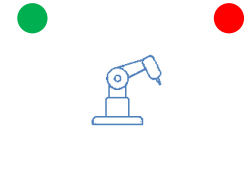
Image: https://pixabay.com/de/idee-antwort-erleuchtung-klugheit-1026393/

Content

Overall goal: Acting properly

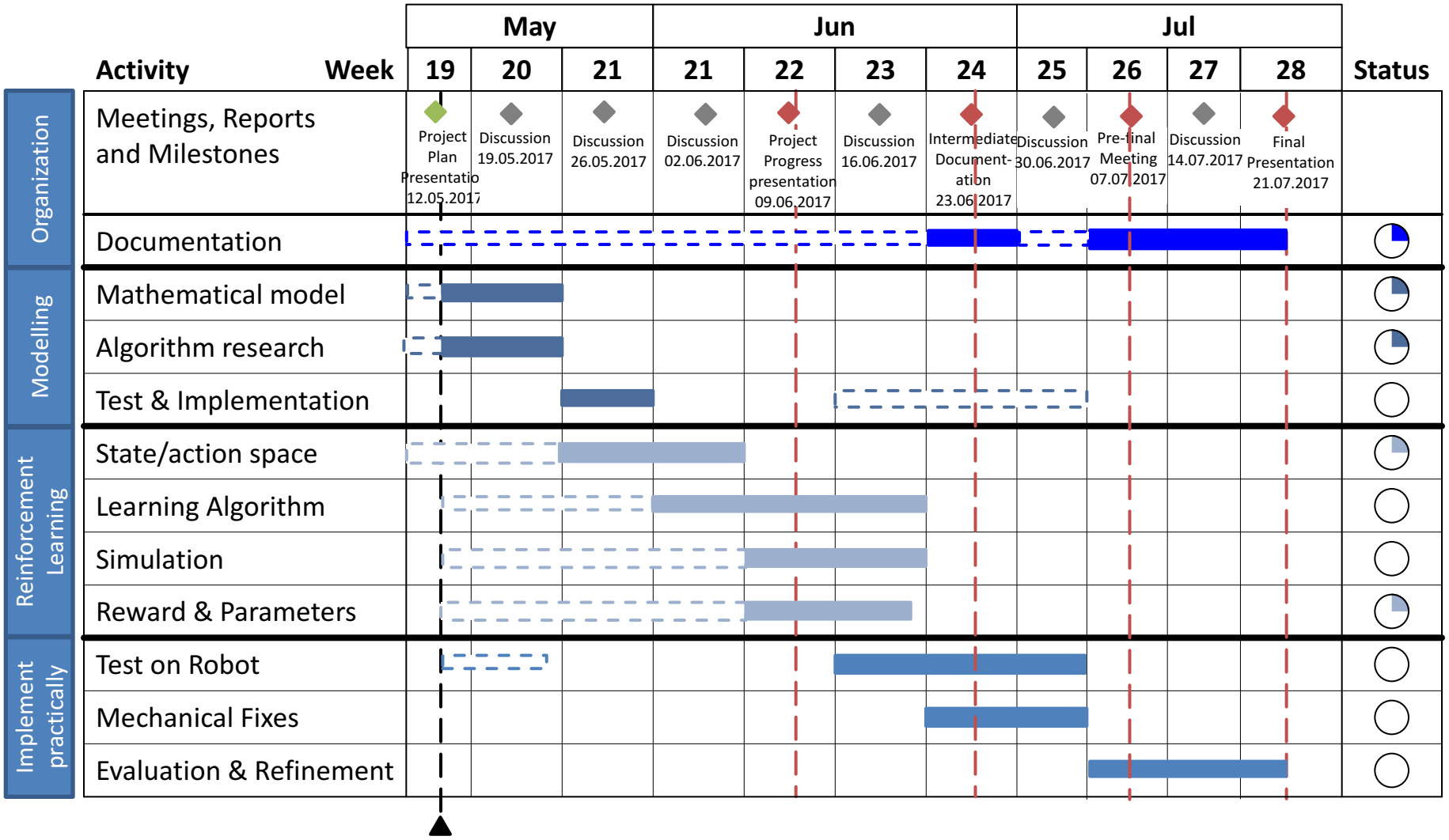
Subgoal	Tasks and Steps	

Mathematical Model and object localization	**Modelling Task**
	▪ Construction and Transformation of world coordinate to relative coordinate
	▪ Algorithm research
	  ▪ Localization through pseudo-localization algorithm
	  ▪ Create the penalty schema
	▪ Algorithm Testing and Implementation
		

Finding optimal policy	**Reinforcement Learning Task** ▪ State and action space formulation ▪ Reward design ▪ Learning Algorithm and Implementation ▪ Simulation (Mathematical model + Nengo)	
Making Robot what the simulator does	**Practical Implementation** ▪ Mechanics and Control ▪ Transfer of Algorithms on real robot arm ▪ Modelling the uncertainty ▪ Performance Evaluation	

Content

Project plan - visualized using a Gantt-Chart



Task distribution

Modelling → Reinforcement Learning → Practical Implementation

Step / Subtasks	Who	Deadline
Simulation Desgin		
▪ Write a python script that makes robot perform predefined movements taking measurement of the environment	W	19/05
▪ Generate the artificial states using the mentioned algorithm	B/M	19/05
ALGORITHM RESEARCH		
▪ Search for distance extraction algorithm and the python APIs (coordinate transformation and relative distance etc.)	B	19/05
▪ Search for the construction and training schema for spiking neural network and a proper reinforcement learning algorithm	M	19/05
TESTING AND IMPLEMENTAION		
▪ Find possibilities to implement these algorithms in python using APIs and libraries or write the code directly.	M/B/W	26/05
▪ Test the algorithms offline using artificial generated states	M/B	26/05

Image: Reference: https://pixabay.com/de/umzug-tragen-schrank-treppe-hoch-1015581/

Task Distribution

Modelling | **Reinforcement Learning** | Practical Implementation

Step / Subtasks	Who	Deadline
STATE AND ACTION SPACE FORMULATION		
▪ Create artificial states that will be used as input data to the simulator (the artificial states can be generated based on the measurement taken with robot and based on geometrical reasoning)	B/W	02/06
LEARNING ALGORITHM AND IMPLEMENTATION		
▪ Create a python default template (i.e. import relevant modules, define spiking neural network and respective default values for actions, states, rewards, value function) to allow for simple coding.	W/B	16/06
▪ Create a function prototype for a function that returns an action according to a policy	M	16/06
MATHEMATICAL SIMULATION		
▪ Try several learning algorithms with the predefined setup and make them run properly on simulator using the artificial states as an input sequence.	B/W/M	16/06
▪ Compare the different learning algorithms based on stability, robustness and speed in the simulator	B/W/M	16/06
REWARD DESIGN		
▪ If necessary do an additional tuning of the rewards or the discount factor using the previously created functions.	B/W/M	16/06

Image: Reference: https://pixabay.com/de/umzug-tragen-schrank-treppe-hoch-1015581/

Task Distribution

Modelling → Reinforcement Learning → **Practical Implementation**

Step / Subtasks	Who	Deadline
TRANSFER OF ALGORITHMS ON ROBOT		
▪ Identify possible mechanical issues and create fixes (e.g. check the movement)	B	30.06
MECHANICS AND CONTROL		
▪ Implement algorithm prototypes (Spiking neural network based Reinforcement learning) on robot and identify, if the performance/behavior largely deviates form the simulator	M/B/W	07.07
PERFORMANCE EVALUATION		
▪ Evaluate the performance and possibly improve the algorithm using simulation and real robot.	M/B/W	21.07

Image: Reference: https://pixabay.com/de/umzug-tragen-schrank-treppe-hoch-1015581/

Task Distribution

Modelling → Reinforcement Learning → Practical Implementation

Step / Subtasks	Who	Deadline
TRANSFER OF ALGORITHMS ON ROBOT		
▪ Identify possible mechanical issues and create fixes (e.g. check the movement)	B	30.06
MECHANICS AND CONTROL		
▪ Implement algorithm prototypes (Spiking neural network based Reinforcement learning) on robot and identify, if the performance/behavior largely deviates form the simulator	M/B/W	07.07
PERFORMANCE EVALUATION		
▪ Evaluate the performance and possibly improve the algorithm using simulation and real robot.	M/B/W	21.07

Documentation will be handled by all group members continuously but mainly from 01/07 to 14/07

Image: Reference: https://pixabay.com/de/umzug-tragen-schrank-treppe-hoch-1015581/

BACKUP

Module: Modelling

BACKUP

Recent Results

- Result 1
- Result 2
- Result 3
- Result 4
- Result 5

Risks

- Risk 1
- Risk 2
- Risk 3
- Risk 4
- Risk 5

Upcoming Tasks

- Task 1
- Task 2
- Task 3
- Task 4
- Task 5

Subgoal / Step	
Modelling Goal 1	
Modelling Goal 1	
Modelling Goal 1	
Modelling Goal 1	
Modelling Goal 1