THE UNIVERSITY OF SHEFFIELD
Department of Electronic and Electrical Engineering

MSc Individual Project

Interim Report

Student Name	Yuche Huang		
Project Title	Using reinforcement learning for balancing double pendulum		
Supervisor	Dr. Peter Rockett	**Second Marker**	Dr. Mark Hopkinson

Description and aims of Project:

Reinforcement learning (RL) is a subfield of machine learning which offers the ability to enhance future manipulation in a dynamic system by using previous experience via interacting with the environment. Due to some of unknown states or actions are difficult to describe by existed control models, designing a control system might be puzzled. Therefore, RL plays a crucial role to alleviate the issue by evaluating the performance of using vastly different approaches operating in the same circumstances to find the best sequence of inputs operating in a dynamical system and optimise the performance based on fundamental knowledge of the respond caused by input [1]. *Currency, the technique of reinforcement learning has been employed into various filed to adapt a diverse environment and alleviate the complex problems. For instance, the game of Go chess is one of the hardest chess because of the decision for each step is determined by not only considering the current circumstance but predicting the forward situation due to the diversity of the environment. As a result, it is a conundrum to determine the optimise reward based on the rules by using the existed control models for the Go chess-playing algorithm. However, by using RL, the algorithm evaluates the reward given by training significant rounds of episodes to offers the ability to make a optimise the decision by considering the reward in the long term. The performance of the algorithm is demonstrated to beat the best human player in a game of Go in 2016* [2].

According to contribution offered by RL that has been mentioned above, the technique has been researched and implemented by three common methodologies, including deep learning, Q-learning and genetic programming. Each method has different efficiency of adapting the environment affected by parameter. In addition, Markov decision process (MDP) is employed as a concept to achieve RL algorithm by modelling the interaction between the agent and its environment as the finite state machine, and the model is illustrated in fig. 1. The principle is shown as following: agent makes the corresponding action that depends on the current state given from the environment. Simultaneously, the environment responds a new status affected by the action determined by the agent and generates reward or punishment as an evaluation of decision quality [3].

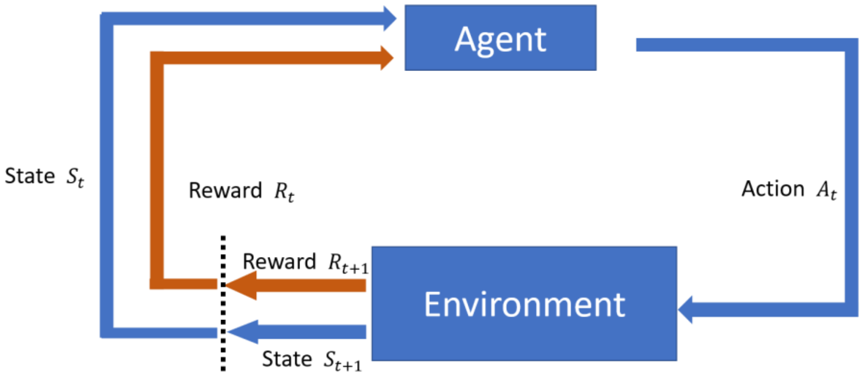


Fig. 1 the structure of Markov decision processes

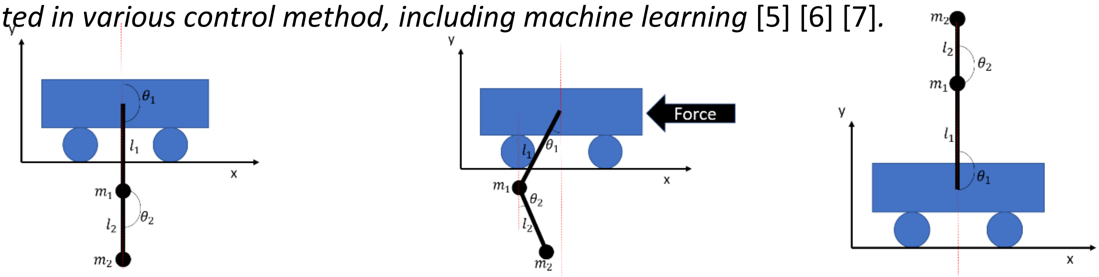
In addition, due to the limitation of sensors set, they might not always offer adequate information for representing the current state that offered by the environment, these hidden states may result in the algorithm controlling unproperly and determining an inaccurate decision [4]. *Therefore, the aim of the project is analysing the performance which adopts the non-Markov decision process that the algorithm generates the action by not only the current states but previous states and action as explicit inputs. Moreover, the research evaluates whether exploiting non-Markov decision process contributes a better performance by comparing with MDP which applied in both Q-learning and genetic programming to solve the pendulum balancing problem. In addition, to simulate the dynamic problem, the work employees the double pendulum as the environment for algorithm training, and analysis the control performance by considering the different number of previous explicit inputs.*

Background theory:

In order to establish a dynamic environment, the movement of the double pendulum is exploited by the work due to it is one of a classic nonlinear and dynamic problem with a strong sensitivity to the initial status. Based on the characteristic, it has been employed in the research to deriving the control equation by using various methodologies such as Q-Learning, genetic programming and Deep learning, in the field of reinforcement learning. In this following section, the background of the research is separated into three topics, including deriving the different equation of double pendulum movement, genetic programming and Q-learning.

-Different equation of double pendulum movement

The double pendulum is a classic mechanical system with an extremely unpredictable behaviour which consists of two links that a pendulum connected by a cart and ended by a second one on the other side. The cart is given two directions of external force to actuated links act as a passive double pendulum. The main purpose of the controller system is to swing up the double pendulum and balance form the given initial status that both pendulums directly point to the ground. The behaviour of Due to the nonlinear characteristic, inverting a double pendulum has been a famous topic that has been adopted in various control method, including machine learning [5] [6] [7].



(a) Initial status *(b) Swing up* *(c) Balance*

Fig. 2 Behaviour of Cart double pendulum balancing

In order to determine the element of state observed form the environment, the state vector of the pendulum system can be represented in x=($\theta_1, \theta_2, \theta_1', \theta_2', \theta_1'', \theta_2''$), where , θ_1, θ_1' and θ_1'' are the angle, angular velocity and the angular accelerator of each pendulum. To simulate the movement of the pendulum, the directly methodology is to identify the equation of angular accelerator in each pendulum by using LaGrange formula. Firstly, by assuming the coordinate of the cart is (0,0), the position and velocities of each pendulum can be represented into the following formula:

Positions:

$$x_1 = l_1 \sin\theta_1$$
$$y_1 = -l_1 \cos\theta_1$$
$$x_2 = x_1 + l_2 \sin\theta_2 = l_1 \sin\theta_1 + l_2 \sin\theta_2$$
$$y_2 = y_1 - l_2 \cos\theta_2 = -l_1 \cos\theta_1 - l_2 \cos\theta_2$$

Velocities:

$$x_1' = l_1 \theta_1' \cos\theta_1$$
$$y_1' = l_1 \theta_1' \sin\theta_1$$
$$x_2' = x_1' + l_2 \theta_2' \cos\theta_2 = l_1 \theta_1' \cos\theta_1 + l_2 \theta_2' \cos\theta_2$$
$$y_2' = y_1' + l_2 \theta_2' \sin\theta_2 = l_1 \theta_1' \sin\theta_1 + l_2 \theta_2' \sin\theta_2$$

According to the paper [6]*, the Euler-Lagrange equation can be demonstrated as* $L = T\ - U$*, where T is the total kinetic energy of the double pendulum according to the speed and mass of the object and U represents the potential energy of the pendulums affected by the altitude comparing with the zero-energy point. These equations are illustrated as shown as following.*

$$T = \frac{1}{2}m_1\left({x_1'}^2 + {y_1'}^2\right) + \frac{1}{2}m_2({x_2'}^2 + {y_2'}^2)$$

$$U = g(m_1y_1 + m_2y_2)$$

Substituting the velocities and the pendulum position into the kinetic energy equation and potential energy formula, then

$$T = \frac{1}{2}(m_1 + m_2)l_1^2{\theta_1'}^2 + \frac{1}{2}m_2l_2^2{\theta_2'}^2 + m_2l_1l_2\theta_1'\theta_2'\cos(\theta_1 - \theta_2) \qquad (1)$$

$$U = -g[(m_1 + m_2)l_1\cos\theta_1 + m_2l_2\cos\theta_2] \qquad (2)$$

Base on the formula above, the Euler-Lagrange equation becomes

$$L = \frac{1}{2}(m_1 + m_2)l_1^2{\theta_1'}^2 + \frac{1}{2}m_2l_2^2{\theta_2'}^2 + m_2l_1l_2\theta_1'\theta_2'\cos(\theta_1 - \theta_2) + g[(m_1 + m_2)l_1\cos\theta_1 + m_2l_2\cos\theta_2] \quad (3)$$

The result of the Lagrange equation is exploited to derive the equation of motion and is demonstrated by using generalised coordinates as shown (4), and the motion formulas are determined separately by each angle variable and shown as the following equation (5) and (6).

$$\frac{d}{dt}\left(\frac{\delta L}{\delta\theta_i'}\right) - \frac{\delta L}{\delta\theta_i} = 0 \qquad (4)$$

For θ_1 *the Euler-Lagrange equation is shown as following.*

$$\frac{d}{dt}\left(\frac{\delta L}{\delta\theta_1'}\right) = \frac{d}{dt}[(m_1 + m_2)l_1^2\theta_1' + m_2l_1l_2\theta_2'\cos(\theta_1 - \theta_2)]$$

$$= (m_1 + m_2)l_1^2\theta_1'' + m_2l_1l_2[\theta_2''\cos(\theta_1 - \theta_2) - \theta_2'(\theta_1' - \theta_2')\sin(\theta_1 - \theta_2)] = 0 \qquad (5)$$

Similarly, by calculating the same procedure, the equation for θ_2 *is derived as shown.*

$$\frac{d}{dt}\left(\frac{\delta L}{\delta\theta_2'}\right) - \frac{\delta L}{\delta\theta_2'} = 0$$

$$= m_2l_2\theta_2'' + m_2l_1\theta_1''\cos(\theta_1 - \theta_2) - m_2l_1{\theta_1'}^2\sin(\theta_1 - \theta_2) + m_2gsin(\theta_2) = 0 \qquad (6)$$

In order to solve the formulas (5) and (6) numerically, the aim of the theory is to establish the relationship of angular accelerator and other parameters. By rearranging these equations, the final formula is demonstrated below.

$$\theta_1' = \frac{d\theta_1}{dt}$$

$$\theta_2' = \frac{d\theta_2}{dt}$$

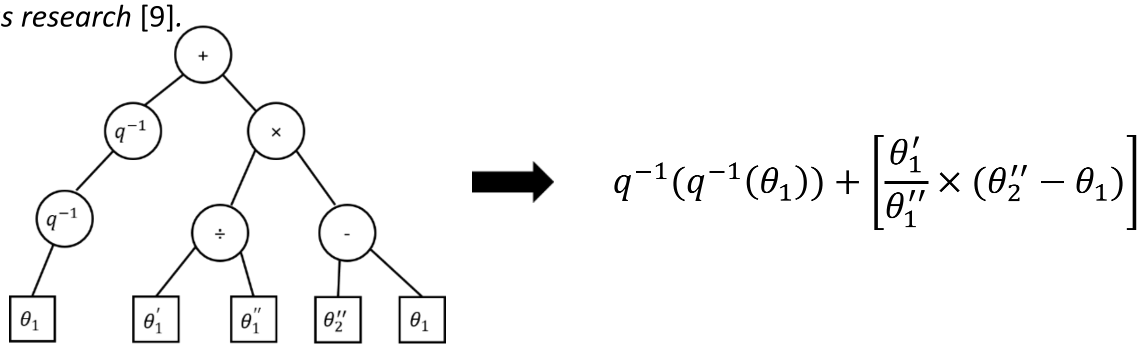
$$\theta_1'' = \frac{m_2l_1{\theta_1'}^2\sin\big(2(\theta_1 - \theta_2)\big) + 2m_2l_2{\theta_2'}^2\sin(\theta_1 - \theta_2) + 2gm_2\cos\theta_2\sin(\theta_1 - \theta_2) + 2\mathrm{g}m_1\sin\theta_1}{-2l_1[m_1 + m_2sin^2(\theta_1 - \theta_2)]}$$

$$\theta_1'' = \frac{m_2l_2{\theta_2'}^2\sin\big(2(\theta_1 - \theta_2)\big) + 2(m_1 + m_2)l_1{\theta_1'}^2\sin(\theta_1 - \theta_2) + 2\mathrm{g}(m_1 + m_2)\cos\theta_1\sin(\theta_1 - \theta_2)}{2l_1[m_1 + m_2sin^2(\theta_1 - \theta_2)]} \qquad (7)$$

-Genetic programming

Genetic programming is a concept to allow the algorithm to adapt the environment based on the evolution theory demonstrated by Darwin: species modify their action by interacting with the environment in order to survive and increase [8]. *Based on the theory, genetic programming adopts a considerable candidate sequence control functions as the sample to search for approximate solution by evaluating via the fitness function to simulate the phenomenon of evolution in the algorithm. To analysis the concept of genetic programming, this evolutionary computation can be divided into three sections including: initialize, creating generation and evaluation.*

Initialization is the first step of the evolutionary algorithm which determines the critical parameter: maximum tree depth. The parameter is adopted for restricting the size of the initialized tree and consisted of two primary sets that including possible of terminals T and internal nodes I where T usually represent input value such as θ_1', θ_1'' or other constant value, and I implies the operator of cothe ntrol function. The example of the tree-based GP is illustrated as fig. 3(a) where the circle represents the operator as a node and square implies the input parameter as the terminal node. In addition, based on the tree structure, the policy of controlling the dynamic system is illustrated as the formula as fig. 3(b). Moreover, the structure of tree-based GP can be assembled by two common methodologies: grow method and full method, and the algorithm of these method is identified in the previous research [9].

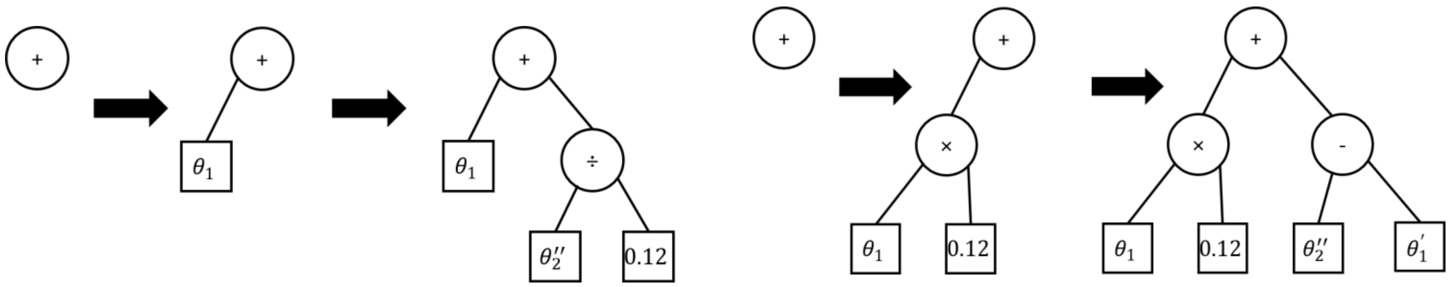


$$q^{-1}(q^{-1}(\theta_1)) + \left[\frac{\theta_1'}{\theta_1''} \times (\theta_2'' - \theta_1)\right]$$

(a) Example of tree-based GP *(b) the candidate control formula*

Fig. 3 Example of genetic programming candidate solver

The grow method can be implemented by using recursive sub-function to return a sub-node or terminal within the given depth of the tree size. For instance, the root node creates a terminal child node while generating the second child node which becomes another internal node containing another terminal node. Comparing with grow method, full method adopting the similar concept to extend the node but with an extra characteristic. If the depth of the tree node is smaller than the maximum size, it allows the child node being generated by choosing from the set of operators randomly rather than ending up by the terminal nodes before exceeding the size limitation. Both methodologies for assembling the tree structure are illustrated as shown as following fig. 4.

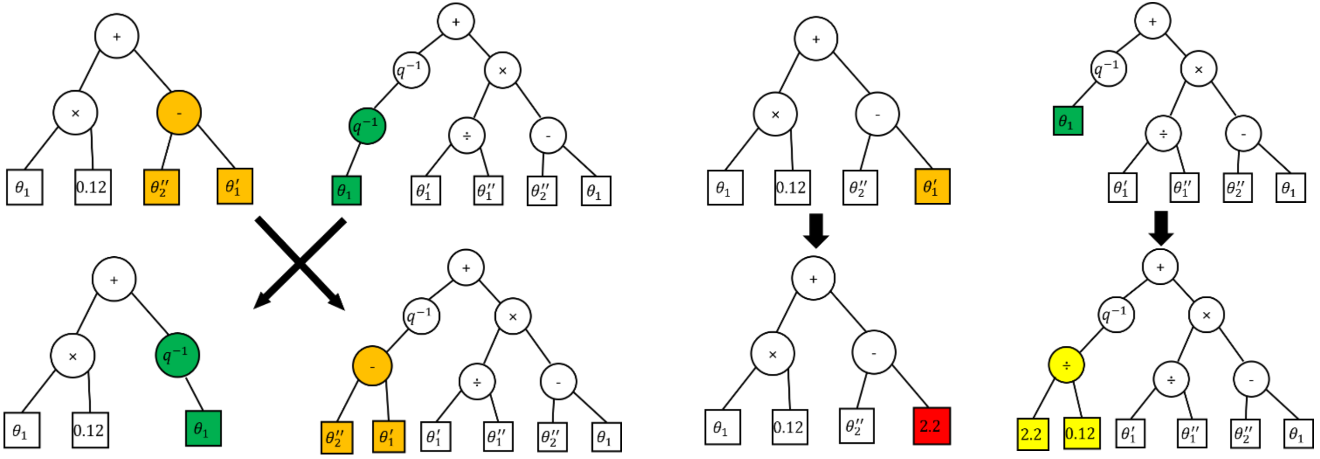


(a) grow method *(b) full method*

Fig. 4 Two common methods for constructing GP structure tree

Furthermore, in order to diverse the candidate solutions and identify the approximate control formula, the genetic operators are employed in this algorithm to generate the next generation which keeps some of characteristic form parent's candidate [10]. *After evaluating the candidate solutions by using fitness function, the next generation is created by selecting the parent with higher reward form the current population. These termed solutions are adopted to produced offspring by using the following genetic operator: crossover and mutation.*

The crossover is implemented by exchanging genetic material which is the subsection of the parent structure tree, from two or more current candidate solutions. On the other hand, comparing with the crossover, the mutation operator offers a function that only substituting a single individual genetic material by the subtree with a randomly created tree. By using these two genetic operators, it implements the simulation that each solution has a slightly different ability to adapt to the environment from others according to the evolution theory. In addition, these operators allow the algorithm approaches to the term solution slowing in each generation step to find the optimize solution. The principle of each methods is illustrated in the fig. 5.



(a) Crossover

(b) Mutation

Fig. 5 Genetic operator

Moreover, according to the research [11], *in order to evaluate the candidate solutions, the reward function is exploited to statist the reward value to judge the ability the function interacts to the environment. To describe the behaviour of the system, the procedure from initial status to the term status is assembled by several step states and actions. The process of dynamic system movement can be represented into the following equation (8), where S and A are indicated the state and the action in the current timestep. In addition, the reward value is affected by the action determined by the agent, the relationship can be illustrated in formula (9). Due to offering the ability for algorithm considering the further reward, the discount factor ϒ (between 0 to 1) is adopted into the formula that indicates future rewards have a lower value than currency rewards* [12]. *By using the reward function, the total value for using candidate policy π is employed to evaluate whether the function is the optimized solution for the environment.*

$$S_0 \overset{a_0}{\rightarrow} S_1 \overset{a_1}{\rightarrow} S_2 \overset{a_2}{\rightarrow} S_3 \overset{a_3}{\rightarrow} S_4 \ldots \quad (8)$$

$$V^{\pi} = R(S_1, a_0) + \Upsilon R(S_1, a_0) + \Upsilon^2 R(S_1, a_0) + \Upsilon^3 R(S_1, a_0) \ldots \quad (9)$$

Methodology:

The programming is divided into two sections including environment and RL algorithm. In the environment section, originally, the work employees the OpenGL library to establish the environment. Due to OpenCL is developed for 3D plotting, the CImg library is the intuitive method with a compact source document to achieve the aim. In order to simulate the moving of the double pendulum, the CImg library is employed into the project due to it provides the function of plotting animation in C language project achieved by designing the pattern in each frame. Due to the library support various function for plotting, the project adopts only the critical function demonstrated in the library datasheet [13] *in order to minimax the size of the program. Moreover, the procedure of the program is shown as following fig. 6*

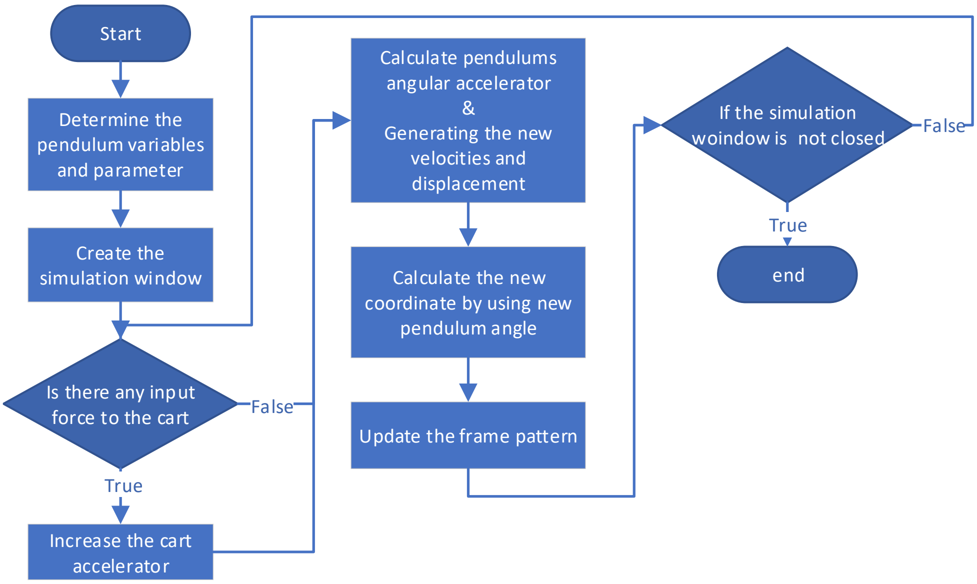

Start
Determine the pendulum variables and parameter
Create the simulation window
Is there any input force to the cart
False
True
Increase the cart accelerator
Calculate pendulums angular accelerator & Generating the new velocities and displacement
Calculate the new coordinate by using new pendulum angle
Update the frame pattern
If the simulation woindow is not closed
False
True
end

Fig. 6 The procedure of double pendulum movement

The main function allows the terminal (operation system) to generate a new window for illustrating the simulation by using CImg library. In addition, the conditional loop which only breaks by closing the simulation window is employed by the program to update the frame by plotting the next pattern in per round. In order to prevent the simulation form enouncing error due to the overflowing variables caused by the finite sequence of input, the friction is used in the program and demonstrated by only considering kinetic friction. Moreover, the damping factor is applied by the project to offer an adjustable environment condition and flexible function when analysing the movement step.

Furthermore, the angular accelerator of each pendulum is generated at the beginning of the loop by using the differential equation derived by the Euler-Lagrange mentioned above. By using the relation among velocity, acceleration and shift, the new coordinate of each pendulum is generated by using the addition previous position with displacement. At the end of the loop, the frame is cleared as an empty paper and replot a new pattern according to the new coordinate to illustrate the movement of the pendulum. In order to analysis the movement of pendulum flexibly, the waiting function is adopted for offering an adjustable frame updating rate.

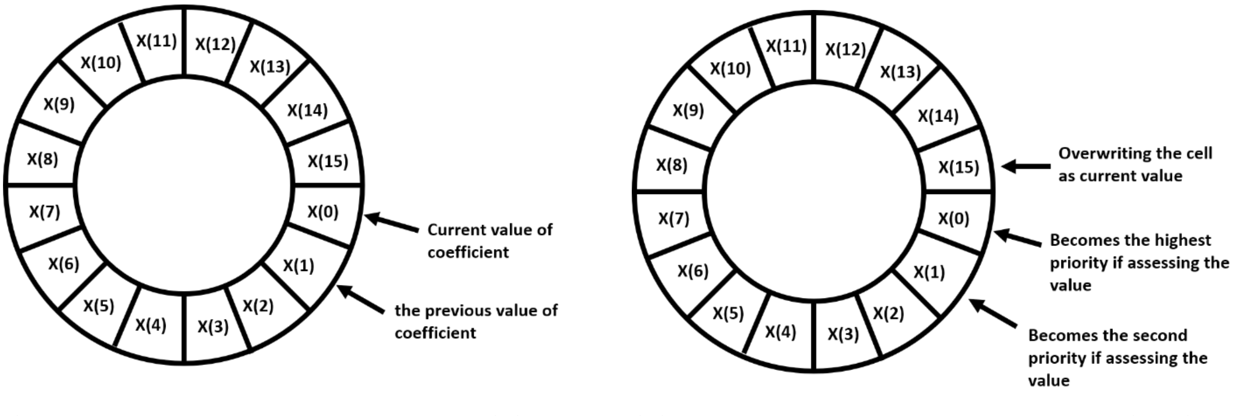
In the algorithm section, genetic programming and Q-learning are employed to adapt the environment to maximize the reward given by the acceptable angular and position taken by the agent in each step. To achieve the aim of the project, the work evaluates the control performance by adopting the different number of past experiences as the explicit inputs while setting other parameters such as generations, mutation, crossover and operator as constant values in each experiment. These parameters are assumed as the following table. 1. Furthermore, the parameter of the pendulum system is determined by referring to the previous paper and shown as the table. 2. In order to achieve the function of assessing the past value, the ring registers are employed by the project due to it offers the function of storing the value of each parameter in per step, and the concept of the ring register is demonstrated as following fig. 7.

Parameter	Value
Fitness:	Reward
Generations:	300
Population size:	500
Selection:	Tournament
Mutation:	15%
Crossover:	85%
Operator set:	{ Move to left, Move to right, +, -,×, ÷ ,q^{-1}, q^{-2}, q^{-3}, AQ [14] }
Terminal set:	{ θ_1, θ_2, θ_2'}

Table. 1 Genetic programming parameters

Parameter	Symbol	Value
Mass 1	m_1	1 kg
Mass 2	m_2	1 kg
Length of link 1	l_1	1 m
Length of link 2	l_2	1 m
Time step	t	0.01 us
Gravitation	g	9.8

Table. 2 The pendulum parameters value


X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)
X(8)
X(9)
X(10)
X(11)
X(12)
X(13)
X(14)
X(15)
Current value of coefficient
the previous value of coefficient
Overwriting the cell as current value
Becomes the highest priority if assessing the value
Becomes the second priority if assessing the value

(a) the current storing location when t(n) (b) the current storing location when t(n+1)

Fig. 7 The structure of the ring register

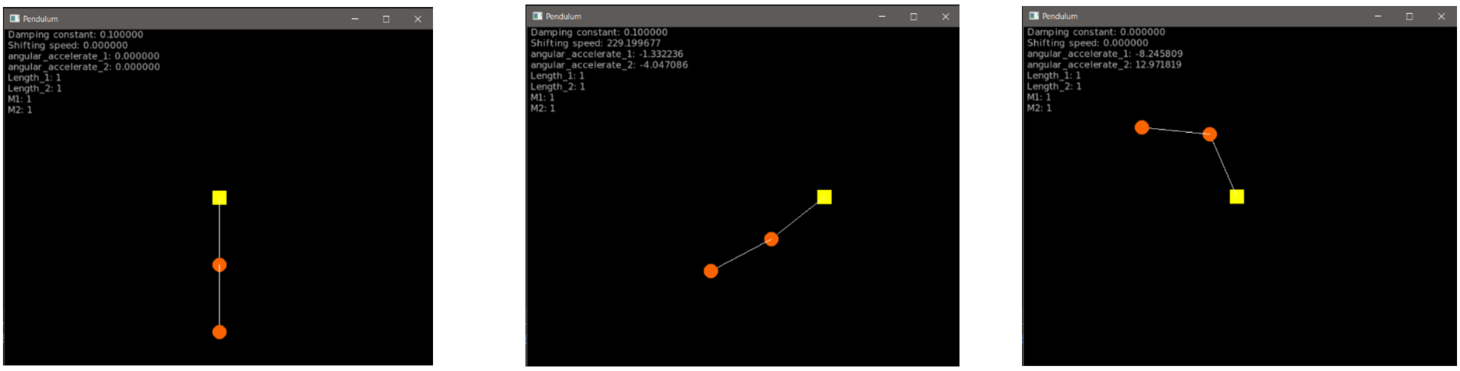
Traditionally, shift registers are exploited for storing the past information that offer the intuitive concept that the current value is always stored in the first element while shifting the remain of the data. However, by using the method, it consumes a considerable number of machine cycle that highly depends on the length of the array to achieve the function. Thus, to minimax the loading for the computer, the ring register is adopted in this project to alleviate the issue. The ring register can be implemented as the following concept. By shifting the storing index, fig. 7(b), the oldest element, x(15), is overwritten as the current value while its previous neighbour index, x(0), becomes the highest priority location while assessing the previous value. Based on the concept, the operation that storing the massive past parameter value, is decreased with only one instruction cycle and no longer affected by the length of the array anymore.

Result and analysis:

Originally, the movement of the double pendulum is hard to simulate properly due to the displacement, external force and friction factor are difficult to identify in a suitable value. By testing considerable experiment, the optimal parameter is determined and illustrated in table. 3. Moreover, in order to test the function of balancing the double pendulum, the external force is given to derive the system by pressing the left and right key button. In addition, by using the up and down key buttons, it provides the function for adjusting the damping factor in order to provide a fixable simulation. The movement of double pendulum works perfectly as the training environment that demonstrated in fig. 8.

Parameter	*Value*
External force	*6*
Dynamic friction	*0.6*
Frame rate coefficient	*5*
Length of link 2	*1 m*

Table. 3 environment simulation parameters



```
Damping constant: 0.100000
Shifting speed: 0.000000
angular_accelerate_1: 0.000000
angular_accelerate_2: 0.000000
Length_1: 1
Length_2: 1
M1: 1
M2: 1
```

```
Damping constant: 0.100000
Shifting speed: 229.199677
angular_accelerate_1: -1.332236
angular_accelerate_2: -4.047086
Length_1: 1
Length_2: 1
M1: 1
M2: 1
```

```
Damping constant: 0.000000
Shifting speed: 0.000000
angular_accelerate_1: -8.245809
angular_accelerate_2: 12.971819
Length_1: 1
Length_2: 1
M1: 1
M2: 1
```

Fig. 8 Simulation of double pendulum in different state

In order to ensure the simulation working correctly, the work offers another window to record the position of the second pendulum in every time step. By tracing the simulation for a period time, the pattern of the pendulum is compared with the document offered by the Matlab company [15] *and poofs the simulation is correct.*

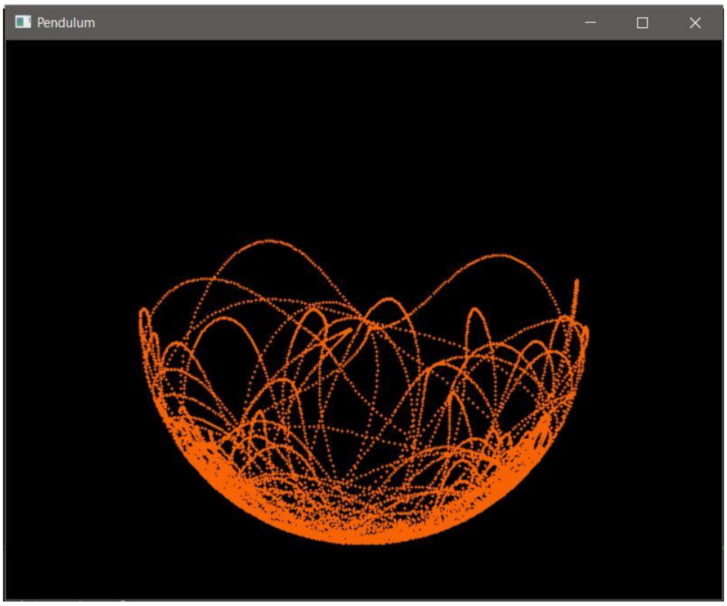


Fig. 9 The trace of double pendulum

Discussion and conclusions:

The result of this work offers the methodology for the developer to improve the control performance. By considering the previous input and output as explicit inputs, the algorithm provides an accurate policy function demonstrated by tree decision generated by genetic programming. Due to the progress delay, the work has not identified the important result relative to genetic programming. By rearranging the Gantt chart, the key result can be determined before July.

Project Specification:

Due to the project will be applied in the embedded system, the project is designed in C language in order to accommodate other applications without any issue. In order to simulate the movement of double pendulum, the differential equation that using the standard methodology of Euler-LaGrange mechanics [6] *is adopted in the project. Furthermore, to minimize the size of the program, the CImg external library is employed in this project to illustrate the simulation of the movement. In addition, the research exploits friction and damping factor to prevent the calculation of double pendulum movement from occurring error due to the variable overflowing such as accelerate and velocity, caused by positive feedback from the equation. To alleviate the issue, the value of and damping parameter is defined as 0.1 according to the previous research* [16]. *Moreover, due to the aim of project concerns lagged variables which using the previous input and output as explicit inputs to generate the decision tree by using genetic programming, Markov decision processes which the agent determines the next action by considering only the current state, is not acceptable in this project. To achieve the non-Markov decision process, the ring registers are adopted to store each value of parameters in genetic programming. By updating the last element in the array, this concept contributes higher operating efficiency comparing with the traditional shift register due to the latter always updates the first element while shifting the rest of the elements by taking up longer machine cycle.*

Milestone evaluation and future work:

The progress of the project is divided into three stages: environment simulation, RL algorithm design & improvement and the IEEE final report writing that is represented in orange, green and blue colour in the Gantt chart. The environment simulation is estimated to be completed at the end of April and start designing the RL algorithm before June. Due to it is the first time to use the CImg library to simulate the movement of double pendulum in the C program, it took approximately one week to read the instructions of the function that the project requires from the library datasheet and caused a week delay for other following tasks. In addition, the issue of simulating double pendulum which needs the adoption of another factor such as friction and damping, is deferred to improve due to the lecture assignments and the preparation for the examination. This leads to the delay in the schedule about designing the algorithm. For the overall progress, the progress has been done so far is slightly delayed comparing with the expected schedule. Therefore, the Gantt chart is rearranged for the rest of the project tasks and the time each task occupies have been re-evaluated. Due to there is no lecture in the third semester, the tasks can be achieved more efficiency than the previous semester. To ensure the time is adequate for IEEE report writing, the task is extended for two extra weeks. For RL algorithm designing, meeting with the supervisor regularly guarantees the path of the project is correct and wasting time on the work which is not relevant to the project is avoided.

Modified Gantt Chart:

A – Background research on RL
B – Project initialisation document
C – Study the theory of movement in pendulum
D – Learning CImg library in C
E – Simulate the movement of pendulum in C
F – Second marker meeting
G – Interim Report
H – Reinforcement learning algorithm design (using C)
I – Evaluation and performance improvement
J – Second marked viva
K – IEEE style Report

Month	Feb		March				April				May					Jun				July					August			
task\Week	1	2	3	4	5	6	7	8	9	10	11	Exam Preparing		Examination Week			17	18	19	20	21	22	23	24	25	26	27	28
A	🟧╲	🟧╲	🟧╲	🟧╲	🟧╲	🟧╲											🟧╲	🟧	🟧	🟧								
B	🟧╲	🟧╲																										
C			🟧╲	🟧╲																								
D				🟧╲	🟧╲	🟧╲																						
E					🟧╲	🟧╲	🟧╲	🟧╲	🟧╲																			
F						🟧╲																						
G																	🟧╲	🟧										
H									🟩╲	🟩╲	🟩╲						🟩╲	🟩	🟩	🟩	🟩							
I																				🟩	🟩	🟩						
J																						🟦	🟦	🟦	🟦	🟦	🟦	🟦
K																							🟦	🟦	🟦	🟦	🟦	🟦

Risk Register:

Identify the key problems that could prevent your project from completing on time and associate a likeliness and risk level (Low/Medium/High). How can these risks be reduced?

Risk Number	Description of Risk	Mitigation of Risk	Risk evaluation (L/M/H)	Chance of risk occurring (L/M/H)	
1	Loss of data (USB key)	Multiple back-ups in multiple locations	M	L	
2	Laptop is broken	Synchronise the file to google driver	M	L	

References:

[1] B. Recht, “A Tour of Reinforcement Learning: The View from Continuous Control,” pp. 1–28, 2018.

[2] J. X. Chen, “The Evolution of Computing: AlphaGo,” *Comput. Sci. Eng.*, vol. 18, no. August 2016, pp. 4–7.

[3] R. Ortner, “Adaptive aggregation for reinforcement learning in average reward Markov decision processes,” *Ann. Oper. Res.*, vol. 208, no. 1, pp. 321–336, 2013.

[4] S. D. Whitehead and L. J. Lin, “Reinforcement learning of non-Markov decision processes,” *Artif. Intell.*, vol. 73, no. 1–2, pp. 271–306, 1995.

[5] T. Liu, C. Chen, and Z. Li, “Method of Inequalities-based Multiobjective Genetic Algorithm for Optimizing a Cart-double-pendulum System,” vol. 06, no. February, pp. 29–37, 2009.

[6] J. Chen and J. Chen, “Chaos from simplicity : an introduction to the double pendulum.”

[7] A. Bogdanov, “Optimal Control of a Double Inverted Pendulum on a Cart,” 2004.

[8] M. Pagel, “Darwin ’ s evolution Marseillaise,” *October*, 1999.

[9] S. Whiteson, M. E. Taylor, and P. Stone, “Empirical Studies in Action Selection with Reinforcement Learning,” vol. 15, pp. 33–50, 2007.

[10] Y.-K. Kim, O.-S. Kwon, Y.-W. Cho, and K.-S. Seo, “Genetic Programming based Illumination Robust and Non-parametric Multi-colors Detection Model,” *J. Korean Inst. Intell. Syst.*, vol. 20, no. 6, pp. 780–785, 2011.

[11] A. Ng, *CS229 Lecture notes 12: Reinforcement Learning and Control*. 2014.

[12] M. Ladeira, “A Comparison Study Between Deep Learning and Genetic Programming Application in Cart Pole Balancing Problem,” *2018 IEEE Congr. Evol. Comput.*, pp. 1–7, 2018.

[13] T. Oct, “The CImg Library 1.3.2,” *Library (Lond).*, pp. 1–4, 2009.

[14] T. Dou, Y. K. Lopes, and P. Rockett, “Model Predictive Control of Buildings Using Genetic Programming Dynamic Models,” no. April, 2019.

[15] “Project 3 - Double Pendulum and Chaos Derivation of Equations of Motion,” vol. 1, pp. 1–12.

[16] E. Physica and U. K. Received, “Dynamics of a parametrically excited double pendulum,” vol. 75, pp. 541–558, 1994.