

---

# 0 Contents

---

|       |                                 |   |
|-------|---------------------------------|---|
| 0.1   | The Inverted Pendulum (IP)      | 2 |
| 0.1.1 | Dynamics                        | 2 |
| 0.1.2 | Cost and Value Function         | 3 |
| 0.1.3 | State Representations           | 3 |
| 0.1.4 | Episode Execution               | 3 |
| 0.2   | Self-Play and Adversaries       | 3 |
| 0.2.1 | Point of Action                 | 3 |
| 0.2.2 | Worst Possible Action           | 3 |
| 0.2.3 | Adversarial Cost                | 3 |
| 0.3   | Neural Network                  | 3 |
| 0.3.1 | Loss Functions and Pareto       | 3 |
| 0.3.2 | Architectures                   | 3 |
| 0.4   | MCTS                            | 4 |
| 0.4.1 | State and Player Representation | 4 |
| 0.4.2 | Terminal States and Suicide***  | 4 |
| 0.4.3 | Modified UCB                    | 4 |
| 0.5   | Player and Adversary Evaluation | 4 |
| 0.5.1 | Elo Scoring                     | 4 |

Explain the assumptions behind the theoretical development you are using and the application of the theory to your particular problem. Any heavy algebra or details of computing work should go into an appendix. This section should describe the running of the experiment or experiments and what equipment was used, but should not be a blow by blow account of your work. Experimental accuracy could be discussed here.

## 0.1 The Inverted Pendulum (IP)

The Inverted Pendulum is an inherently unstable system with highly nonlinear dynamics and is under-actuated.

### 0.1.1 Dynamics

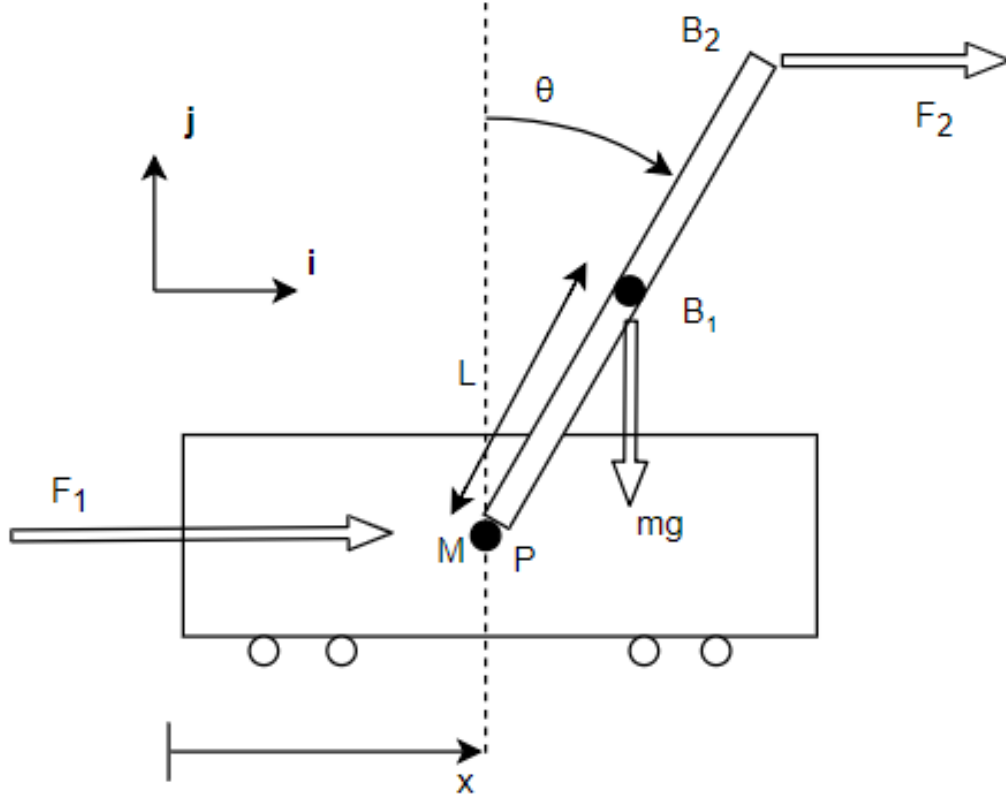


Figure 1: A free-body diagram of the inverted pendulum system. For the OpenAI IP the system is in discrete time with a time-step of  $\tau = 0.02s$ . The other constants are  $l = 0.5m$ ,  $m = 0.1kg$ ,  $M = 1kg$ ,  $F = \pm 10N$ ,  $x_{max} = \pm 2.4m$ ,  $\theta_{max} = \pm 12^\circ$ .

The full state space equations for the inverted pendulum as defined in fig. 1 are given by:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{\left(\frac{2M-m}{m}F_2 - F_1\right)\cos\theta + g(M+m)\sin\theta - mL\dot{\theta}^2\sin\theta\cos\theta}{(M+m\sin^2\theta)} \\ \dot{\theta} \\ \frac{F_1 + F_2\cos(2\theta) + m\sin\theta(L\dot{\theta}^2 - g\cos\theta)}{L(M+m\sin^2\theta)} \end{bmatrix} \quad (1)$$

Ignoring second order terms and linearising about  $\mathbf{x}_e = [x_e, \dot{x}_e, \theta_e, \dot{\theta}_e]^T = [0, 0, 0, 0]^T$ :

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{\frac{2M-m}{m}F_1 - F_2 + g(M+m)\theta}{M} \\ \dot{\theta} \\ \frac{F_1 + F_2 - gm\theta}{lM} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g\frac{M+m}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{mg}{lM} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{M} & \frac{2M-m}{Mm} \\ 0 & 0 \\ \frac{1}{lM} & \frac{1}{lM} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (2)$$

Which, as expected, is unstable since  $\det(\lambda I - A) = 0 \implies \lambda^2(\lambda^2 + \frac{mg}{lM}) = 0$ . For small angles the natural frequency of a non-inverted pendulum is  $\omega_n = \sqrt{\frac{mg}{lM}} = \sqrt{\frac{0.1 \times 9.81}{0.5 \times 1}} \approx 1.40 \text{ rad/s}$ . Therefore, the time constant for the system is  $\tau \approx 0.70 \text{ s}$ . A discrete time step of  $0.02 \text{ s}$  is 35x smaller than this and therefore we expect an impulse to cause  $\sim 3\%$  change in the state values. This is far below the threshold for pulse-width modulation, i.e. the actions are fast enough for the input forces to be modelled as continuous \*\*\*\* Is this right?? experimentally, the largest velocities give even better results \*\*\*\*

## 0.1.2 Cost and Value Function

## 0.1.3 State Representations

pseudocode, proof and explanation + cost/benefits

## 0.1.4 Episode Execution

# 0.2 Self-Play and Adversaries

## 0.2.1 Point of Action

symmetry, action choices and PMW

## 0.2.2 Worst Possible Action

## 0.2.3 Adversarial Cost

representation with the cost function.

# 0.3 Neural Network

## 0.3.1 Loss Functions and Pareto

## 0.3.2 Architectures

Player vs Adversary Architectures? Combined?

## 0.4 MCTS

outline + pseudocode

### 0.4.1 State and Player Representation

### 0.4.2 Terminal States and Suicide\*\*\*

### 0.4.3 Modified UCB

## 0.5 Player and Adversary Evaluation

### 0.5.1 Elo Scoring