Automatic Generation Control for a Two Area Power System Using Deep Reinforcement Learning

Shane Reynolds

Charles Darwin University

Background

A generator's angular acceleration is governed by:

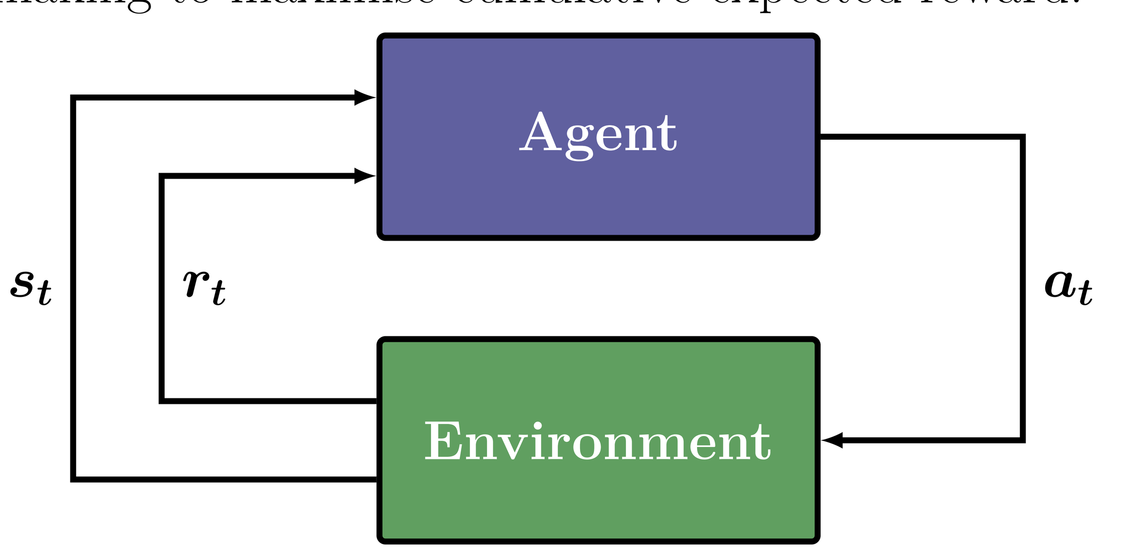
$$\Delta T = T_{mech} - T_{elec} = I\alpha$$

- If $\Delta T > 0$, then $\alpha \uparrow$ and f (Hz) $\uparrow$
- If $\Delta T < 0$, then $\alpha \downarrow$ and f (Hz) $\downarrow$

The Australian power network operates at 50 Hz.

Reinforcement Learning

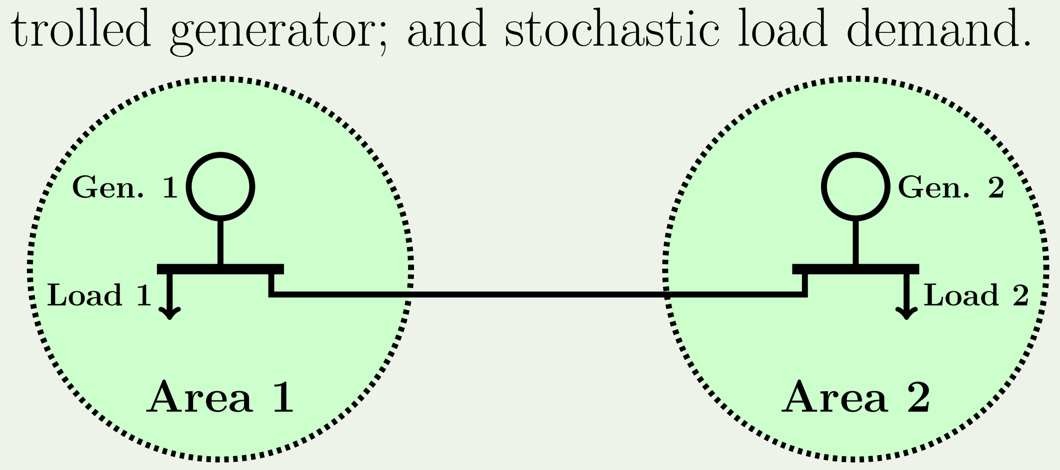
Reinforcement learning is a branch of machine learning concerned with an agent's sequential decision making to maximise cumulative expected reward.



The agent exists in some environment and at each time step observes state $s_t \in S$; and takes an action $a_t \in A$. Following this, the agent then receives a reward $r_t \in R : S \times A \times S \to [R_{min}, R_{max}]$.

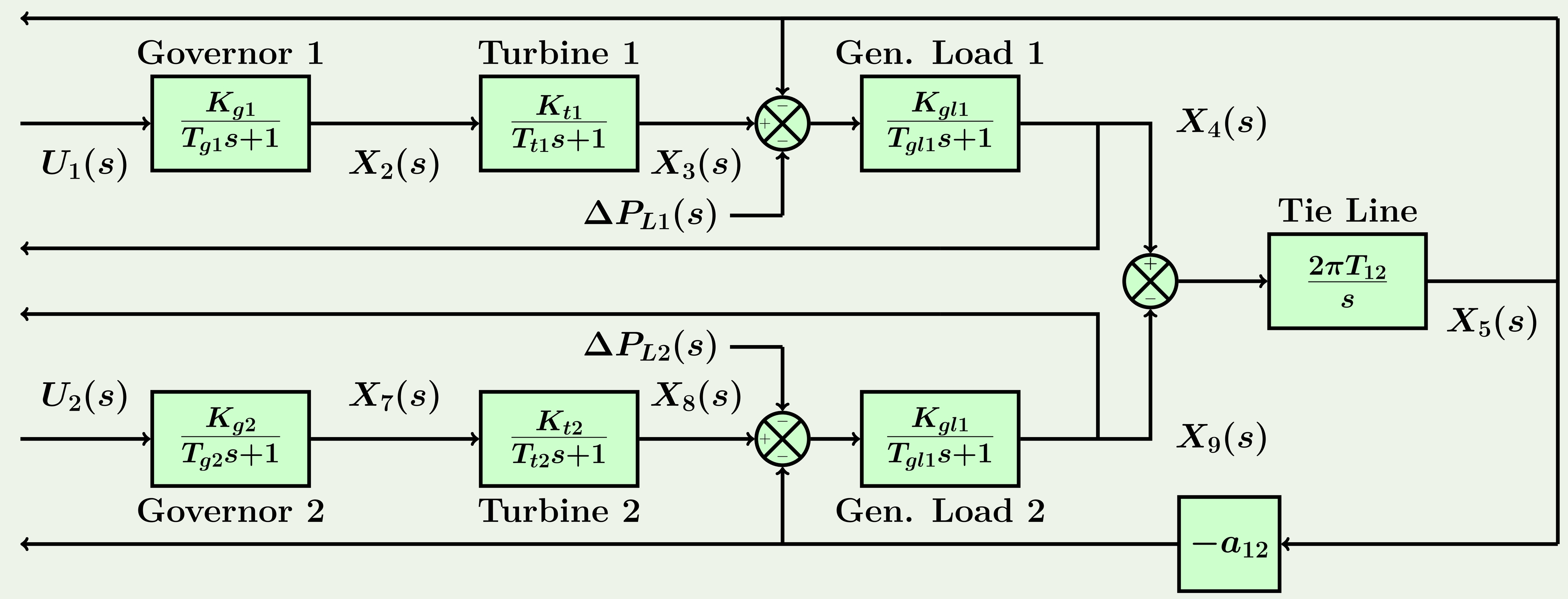
The Environment

Two power areas connected via a transmission line. Each power area consists of: a governor controlled generator; and stochastic load demand.



The control objective is to maintain inter-area power transfer, whilst regulating the frequency of each area.

The Environment in the Frequency & Temporal Domains



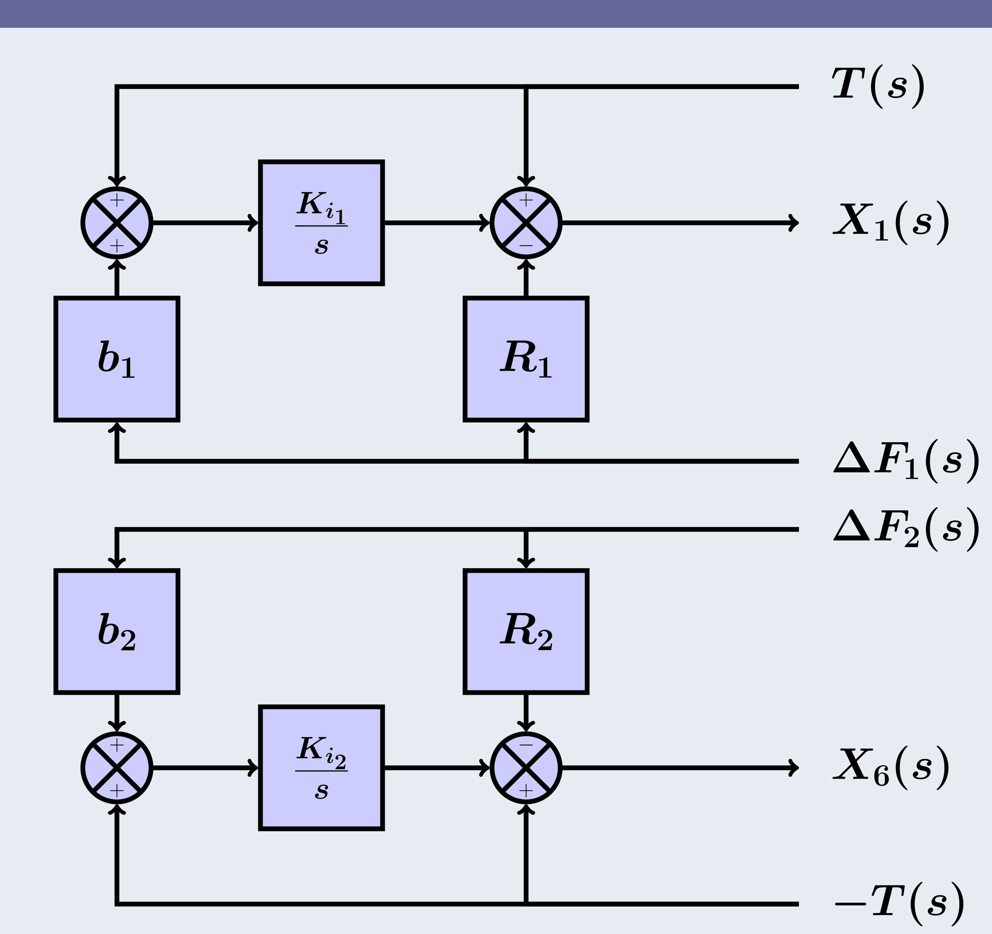
Two Area System ODE

$$\dot{x}_2(t) = \frac{1}{T_{sg_1}}\big(K_{sg_1}u_1(t) - x_2(t)\big) \qquad \dot{x}_7(t) = \frac{1}{T_{sg_2}}\big(K_{sg_2}u_2(t) - x_7(t)\big)$$

$$\dot{x}_3(t) = \frac{1}{T_{t_1}}\big(K_{t_1}x_2(t) - x_3(t)\big) \qquad \dot{x}_8(t) = \frac{1}{T_{t_2}}\big(K_{t_2}x_7(t) - x_8(t)\big)$$

$$\dot{x}_4(t) = \frac{1}{T_{gl_1}}\Big(K_{gl_1}\big(x_3(t) - x_5(t) - \Delta p_{L1}(t)\big) - x_4(t)\Big) \qquad \dot{x}_9(t) = \frac{1}{T_{gl_2}}\Big(K_{gl_2}\big(x_8(t) - x_5(t) - \Delta p_{L2}(t)\big) - x_9(t)\Big)$$

Classical PI Controller



ODE System

$$\dot{x}_1(t) = b_1\Delta f_1(t) + x_5(t)$$
$$\dot{x}_6(t) = b_2\Delta f_2(t) - x_5(t)$$

Experimental Setup

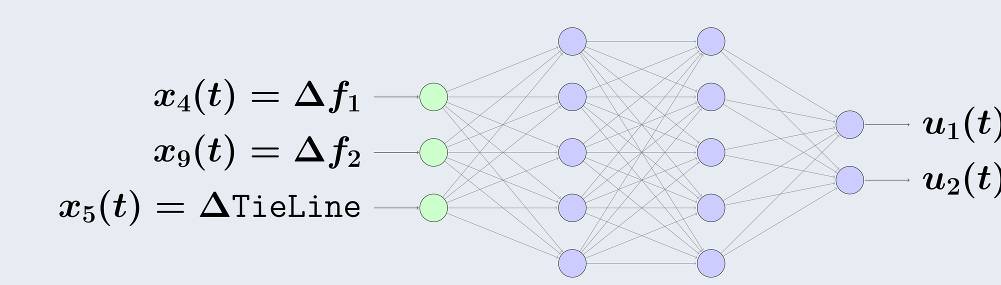
content...

Classical PI Controller Results

content...

DDPG Controller

Neural Network Architecture



Algorithm 1: DDPG

Rand. init. critic $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with θ^Q and θ^μ
Init. target networks Q' and μ' with $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialise replay buffer R
for *episode* $\leftarrow 1$ ***to*** M **do**
  Initialise a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for $t \leftarrow 0$ ***to*** T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ with noise exploration noise
    Execute a_t and observe reward r_t and new state s_{t+1}
    Store transition (s_t,a_t,r_t,s_{t+1}) in R
    Sample random N transitions (s_i,a_i,r_i,s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{Q'}))$
    Update critic by min. loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:
$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_i}$$
    Update the target networks:
$$\theta^{Q'} = \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} = \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$
  end
end

DDPG Controller Results