Action u

Let's encode some square on a board, with a column code (letter a to h), and row code (number 1 to 8) e.g. $e2$, $e4$. Denote all possible codes as set

$$\mathbb{S} = \{a1, a2, \dots, h7, h8\}, |\mathbb{S}| = 64$$

Let's encode an action by a pair of two indices (of beginning square and ending square), e.g.

$$u_k = \langle e2, e4 \rangle \in \mathbb{S}^2$$

Disturbance w

Let's consider Actions of the opponent as a source of Disturbance w, with the same representation as an action of Agent, so

$$w_k \in \mathbb{S}^2$$

Two ways to describe State x:

Since we consider opponent as Disturbance w, we can construct State x, so that it's a sequence of chess boards $\{x_k\}_{k=1}^{\infty}$, before move of our Agent.

x1 — Our move, opponent move — x2 — Our move, opponent move — x3

1. History of all moves and state of timers:

   $$\mathbb{X} = (\mathbb{S}^2 \times \mathbb{S}^2)^T \times \mathbb{R}^2$$

   The dimentionality is not constant, which might be inconvenient.

2. State of the board, timers and controlling boolean flags

   $$x_k = \left\langle B_k, M_k^{agent}, M_k^{opponent} \right\rangle$$

   $$B_k \in \{pawn^b, knight^b, \dots, pawn^w, knight^w, \dots, \varnothing\}^{64}$$

   $$M_k^{agent} = \left\langle t_k, C_a, \bar{C}_h, w_k \right\rangle$$

   $$M_k^{opponent} = \left\langle t_k, C_a, \bar{C}_h, u_{k-1} \right\rangle$$

State space is then:

$$\mathbb{X} = Piece^{64} \times (\mathbb{R} \times \{0,1\}^2 \times \mathbb{S}^2)^2$$

$$|Piece| = 2 \cdot 6 + 1 = 13$$

- 1 timer
- 2 flags per player to control possibility of castling move
- 1 last move of the other to control possibility of en passant move

Output y will be identical to State x:

Notice that chess is a game with complete information. For simplicity let's neglect emotional state and personal goals of players.

This allows us to make State x and Output y identical, thus making State transition law and Output map identical.

$$\mathbb{Y} = \mathbb{X}$$

System type

Even though the State space contains continuous $\mathbb{R}$ values for timers, the law of time transition is very straightforward and there's no harm using sample-and-hold technique to pretend that the State space is discrete.

As we told, every next State x depends on Disturbance w from opponent, and since practically speaking both players are constrained in time to computer their moves, then neither Disturbance w nor Input u are globally optimal definite values, and thus they are random variables.

$$W_{k+1} \sim P_W(w_{k+1}|x_k)$$

Transition PDF

Since every State x depends on previous Disturbance w of opponent, then State x is also a random variable, but calculated as follows:

$$X_{k+1} = f(x_k, w_k, u_k)$$

, where f is a definite function that updates previous state using rules of chess, and it applies opponent's Disturbance w to the board, and then applies Action u of Agent.

Policy k

Given that game of chess is computationally very hard to calculate with an Infinite horizon, in finite time, we can consider policy to be stochastic. Action u is drawn from a PDF (of uknown form), conditioned on the current state (which in turn is conditioned on Disturbance w)

$$U_{k+1} \sim P_U^{\Theta}(u_{k+1}|x_k)$$

If we'd consider chess without time limits, then technically policy can be seen as Markov policy

$$U_{k+1} = \kappa(X_k)$$