Machine and Reinforcment Learning

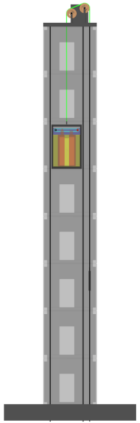
Minimum Time Lift

Lorenzo Rossi - Giacomo Solfizi - Andrea Efficace

Università degli Studi di Roma Tor Vergata

Problema

Il problema del "Minimum Time Lift" consiste nell'utilizzare le tecniche del reinforcement learning per ottenere una policy ottima che permetta il movimento di un ascensore a tempo minimo fra due punti. Una volta ottenuta la policy ottima, deve essere confrontata con la soluzione che si otterrebbe in forma chiusa.



Formulazione

Per risolvere il problema assumiamo che:

- La massa m dell'ascensore è unitaria;
- Le azioni $A_t \in \{-1, 0, 1\}$ per decelerare,accelerare e mantenere l'accelerazione;
- Stati continui e azioni continue;
- Ricompensa -1 per tutti gli istanti di tempo;
- Fine corsa limitati a $-2 \leq y \leq 8$;
- Velocità massima dell'ascensore limitata a $-3 \leq v \leq 3$;

Assunzioni - 1

A livello implementativo abbiamo scelto di seguire la soluzione[1] utilizzata da Matlab che discretizza il sistema e calcola il reward, che possiamo assegnare in due modi:

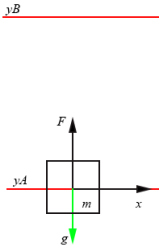
- Reward costanti pari a -1;
- Reward pari all'indice di costo $J(u)$ *(del problema LQR gestito da Matlab)*

[1] Equivalente discreto di un problema LQR

Assunzioni - 2

- Stato iniziale $x_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Stato finale $x_f = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$
- Al lower/upper bound della posizione, accelerazione e velocità nulle e reward -100
- Termine episodio se: $x = x_f$, $v = 0$, $a = 0$
- Forza peso vinta dalla tensione del cavo dell'ascensore

Modello - 1



Sia y_A la posizione iniziale, y_B la posizione finale, $g = 9.81 \frac{m}{s^2}$ accelerazione di gravità e a l'accelerazione impressa sull'ascensore. Allora:

- $F = m * a = m(a - g)$ la forza totale ottenuta dal secondo principio della dinamica;
- $x_1 = X$ posizione della massa;
- $x_2 = \dot{X}$ velocità della massa;
- $u = F$ controllo;

Modello - 2

La dinamica è:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u = Ax + Bu$$

$$y = Cx + Du$$

Algoritmo

- Algoritmo di apprendimento: SARSA(λ) con Elegibility Traces
- Appossimazione funzionale tramite tile coding ($N = 4$, $M = 20$, $k = 2$, $offset = (1, 3)$)
- Scelta dell'azione ε-greedy

Considerazioni

Il comportamento risultante dall'apprendimento varia a seconda del tipo di reward.

Reward costante pari a -1:

Reward pari $J(u)$:

Conclusione

La soluzione trovata tramite reinforcement learning e tramite la risoluzione in forma chiusa del sistema dinamico non differiscono di molto: