Cartpole system

The dynamics of the cartpole system are governed by the following set of differential equations:

$$
\begin{aligned}
\dot{\vartheta} &= \omega \\
\dot{x} &= v_x \\
\dot{\omega} &= \frac{g \sin\vartheta (m_c + m_p) - \cos\vartheta (F + m_p l \omega^2 \sin\vartheta)}{\frac{4l}{3}(m_c + m_p) - l m_p \cos^2\vartheta} \\
\dot{v}_x &= \frac{F + m_p l \omega^2 \sin\vartheta - \frac{3}{8} m_p g \sin(2\vartheta)}{m_c + m_p - \frac{3}{4} m_p \cos^2\vartheta}
\end{aligned}
\tag{1}
$$

where the variables are defined as follows:

- ϑ: pole turning angle (**state variable**) [rad]
- x: x-coordinate of the cart (**state variable**) [m]
- ω: pole angular speed with respect to relative coordinate axes with cart in the origin (**state variable**) [rad/s]
- v_x: absolute speed of the cart (**state variable**) [m/s]
- F: pushing force (**control variable**) [N]
- m_c: mass of the cart [kg]
- m_p: mass of the pole [kg]
- l: pole length [m]

The variable correspondences in the code are as follows:

- ϑ =`angle`
- ω =`angle_vel`
- v_x =`vel`
- m_c =`mass_cart`
- m_p =`mass_pole`

- $l =$length_pole
- $g =$grav_const
- $F =$force

Exercise 1

Theory

Model-predictive control algorithm on every time step t solves the following problem

$$\sum_{k=0}^{H} c(\hat{s}_{t+k\cdot\delta t}, a_k) \to \min_{a_0,\ldots,a_H}, \tag{2}$$

and then applies the first action $a_0^\star$ from the optimized sequence

$$(a_0^\star, \ldots, a_H^\star) = \arg\min \sum_{k=0}^{H} c(\hat{s}_{t+k\cdot\delta t}, a_k)$$

. In the equation (2) above:

- δt is the prediction step size, a tunable hyperparameter which may differ from the system's sampling time.
- $\hat{s}_{t+k\cdot\delta t}$ represents the predicted state at the future time step $t + k \cdot \delta t$. The initial condition is $\hat{s}_t = s_t$, where s_t is the current state vector at time t. The state vector is denoted by $s_t = (\vartheta(t), x(t), \omega(t), v_x(t))^T$. The predicted state is calculated using Euler's method:

  $$\hat{s}_{t+(k+1)\cdot\delta t} = \hat{s}_{t+k\cdot\delta t} + \delta t \cdot f(\hat{s}_{t+k\cdot\delta t}, a_k)$$

  for $k = 0, 1, \ldots, H-1$, where f is the transition function represented by the right-hand side of Equation 1.
- H is the prediction horizon, another hyperparameter
- $c(s,a)$ is the cost function, which is user-defined. For example, $c(s,a) = s^T \operatorname{diag}(w) s$ with w being the vector of cost weights.

Code

Within the code, locate the following function:

```
def mpc_objective(self, current_state, actions) -> float:
```

Implement the function body to compute and return the sum $\sum_{k=0}^{H} c(\hat{s}_{t+k\cdot\delta t}, a_k)$. The variable correspondences are:

- H =`self.prediction_horizon`
- a_k =`actions[k, :]`
- f=`self.system._compute_state_dynamics`
- s_t=`current_state`
- w =`self.cost_weights`
- $c(s, a)$ =`rg.sum(current_state**2 * cost_weights)`
- δt =`self.pred_step_size`