Model Predictive Control and the Unreasonable Effectiveness of Replanning

Sanjiban Choudhury
Landscape of Planning / Control Algorithms

Low-level control

LQR

High-level path planning

LazySP

Figure 2: Solving inverted pendulum swing up using LQR tracking.

Thus far, we have assumed that we were modeling a linear, time-invariant system. As we will see, we might be interested in systems that are linear, but time varying:

\[
\begin{align*}
    x(t+1) &= A_t x(t) + B_t u(t) \\
    c(x(t), u(t)) &= x(t) > Q_t x(t) + u(t) > R_t u(t)
\end{align*}
\]

In this case, the LQR equations are simply updated to:

\[
\begin{align*}
    K_t &= -(B_t V_t + 1 B_t + R_t) V_t \quad (2.3.3) \\
    V_t &= Q_t + K_t R_t K_t + (A_t + B_t K_t) V_t (A_t + B_t K_t) \quad (2.3.4)
\end{align*}
\]

Affine Quadratic Regulation

Let's now consider a generic affine system with time-varying dynamics \( A_t \) and \( B_t \) and a state offset \( x_{off} t \):

\[
\begin{align*}
    x(t+1) &= A_t x(t) + B_t u(t) + x_{off} t \\
    c(\tilde{x}(t), u(t)) &= \tilde{x}(t) > \tilde{Q} \tilde{x}(t) + \tilde{u}(t) > \tilde{R} \tilde{u}(t)
\end{align*}
\]

Affine problems can be converted to linear problems by using homogeneous coordinates:

\[
\begin{align*}
    \tilde{x}(t) &= x(t) + x_{off} t \\
    \tilde{A}_t &= A_t - A_{off} t \\
    \tilde{B}_t &= B_t - B_{off} t \\
    \tilde{Q} &= Q_t + K_t R_t K_t + (A_{off} + B_{off} K_t) \tilde{V}_t (A_{off} + B_{off} K_t) \quad (2.3.7)
\end{align*}
\]

This is just a new LQR problem with modified state and dynamics and a new cost defined as:

\[
    c(\tilde{x}(t), u(t)) = \tilde{x}(t) > \tilde{Q} \tilde{x}(t) + \tilde{u}(t) > \tilde{R} \tilde{u}(t),
\]

where the choice of \( \tilde{Q} \) is problem dependent. We will later see how we can design \( \tilde{Q} \) for the tracking problem. The Affine Quadratic Regulation problem can then be solved in exactly the same way as the LQR problem.

Essentially the same trick can be applied to enable us to have linear cost function terms in the controls as well, but we defer this to the general formulation derived at the end.
Goal: Plan for a real-world helicopter
Takeoff
(Respect power constraints)

Enroute
(Avoid sensed obstacles)

Touchdown
(Plan to multiple sites)
Recap: Solving a MDP

\[ \min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t) \]

(Solve for a sequence of actions)

(Sum over all costs)

\[ s_{t+1} = \mathcal{T}(s_t, a_t) \]

(Transition function)
Brainstorm: Challenges in solving MDP for helicopter

\[
\min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t)
\]

(Solve for a sequence of actions)

(Transition function)

(Sum over all costs)
The Big Challenges

Problem 1: Don’t know the terrain ahead of time!

Problem 2: Don’t have a perfect dynamics model!

Problem 3: Not enough time to plan all the way to the goal!
The Big Challenges

Problem 1: Don’t know the terrain ahead of time!

Problem 2: Don’t have a perfect dynamics model!

Problem 3: Not enough time to plan all the way to the goal!
Activity!
Find a sequence of actions to go from start to goal.

The helicopter can only sense upto 1km.

How should it deal with unknown terrain? What assumptions can it make?
What is the problem mathematically?

\[
\min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t)
\]

(Solve for a sequence of actions)

(Sum over all costs)

\[
s_{t+1} = \mathcal{T}(s_t, a_t)
\]

(Transition function)

Is the transition function fully known?

If not, then how can we solve the optimization problem?
Idea: Plan with an optimistic model

\[
\min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t) \\
\tag{Solve for a sequence of actions}
\]

\[
s_{t+1} = \hat{T}(s_t, a_t) \\
\tag{Sum over all costs)
\]

Assume that any unknown space is fully traversable.

Update model as you get information from real world. Replan!
Plan optimistically and replan as you learn more about the world.
Be Optimistic and Replan!

Stanford DARPA Challenge, 2007
Model Predictive Control (MPC)

Step 1: Solve current MDP (plan) to find a sequence of actions

Step 2: Execute the first action in the real world and update MDP

Step 3: Repeat!
Model Predictive Control (MPC)

Step 1: Solve current MDP (plan) to find a sequence of actions

Step 2: Execute the first action in the real world and update state

Step 3: Repeat!
Model Predictive Control (MPC)

Step 1: Solve current MDP (plan) to find a sequence of actions

Step 2: Execute the first action in the real world and update state

Step 3: Repeat!
The Big Challenges

**Problem 1:** Don’t know the terrain ahead of time!

**Problem 2:** Don’t have a perfect dynamics model!

**Problem 3:** Not enough time to plan all the way to the goal!
Problem 2: Don’t have a perfect dynamics model!

Let’s say there is an unknown gust of wind pushing you off the path.
What is the problem mathematically?

$$\min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t)$$

(Solve for a sequence of actions)

(Sum over all costs)

$$s_{t+1} = \mathcal{T}(s_t, a_t)$$

(Transition function)

Is the transition function fully known?
Problem 2: Don’t have a perfect dynamics model!

Plan with incorrect transition model and replan!

Theorem: An optimal policy in an incorrect model has bounded suboptimality in the real model.
The Big Challenges

Problem 1: Don’t know the terrain ahead of time!

Problem 2: Don’t have a perfect dynamics model!

Problem 3: Not enough time to plan all the way to the goal!
Problem 3: Not enough time to plan all the way to goal!

Example mission:
Fly from Phoenix to Flagstaff as fast as possible (200 km)

Problem:
Take forever to plan at high resolution ALL the way to goal
What is the problem mathematically?

\[ \min_{a_0, \ldots, a_{T-1}} \sum_{t=0}^{T-1} c(s_t, a_t) \]

(Solve for a sequence of actions)

(Sum over all costs)

How large can T be?
What if we planned till a shorter time horizon $T'$?

\[
\min_{a_0, \ldots, a_{T'-1}} \sum_{t=0}^{T'-1} c(s_t, a_t)
\]

(Solve for a sequence of actions)

(Sum over all costs)

Is this even allowed???

Would we get the same solution for $a_0$?
We have to add in a terminal value for the final state

$$\min_{a_0, \ldots, a_{T'-1}} \sum_{t=0}^{T'-1} c(s_t, a_t) + V^*(s_T')$$

(Solve for a sequence of actions)

(Sum over all costs)

(Optimal value of state $s_T'$)

Can we compute the optimal value $V^*$?

If not, how can we approximate it
Idea: Use a global planner to approximate $\hat{V}^*$

$$\min_{a_0, \ldots, a_{T'-1}} \sum_{t=0}^{T'-1} c(s_t, a_t) + V^*(s_{T}')$$

(Solve for a sequence of actions)
(Sum over all costs)
(Approximate value of state $s_{T'}$)

For example: Run a 2D planner from $s_T$ to the goal

Use the cost of that plan to compute approximate value