Control Theory : Course Summary

Author: Joshua Volkmann

Abstract

There are a wide range of problems which involve making decisions over time in the face of uncertainty. Control theory draws from the fields of Engineering, Operations research and Mathematics to supply tools to regulate or optimise a system. I will briefly introduce and discuss some relevant topics considered as part of control theory. Phenomena can be modelled as dynamical systems, where there is some level of control that can be imposed on the system. This includes process control, optimal stopping, automated planning and general strategic decision making in the face of uncertainty.
In these problems there exists some optimal behaviour or optimal policy that is desired.

Markov Decision Problems

An important class of characterising problems are Markov decision problems, which are equivalently known as Markov decision processes (MDPs). Essentially most real-life stochastic control problems can be posed as an MDP. This class of problems is backed by an elegant theoretical framework and is underpinned by Bellman's optimality equation (1), which can be used to solve Markov decision processes.

Problem Definition

To solve a control application it is important to have well-defined problems. In this regard, the concept of states in a state space is important, where the state space is the set of all states that could be visited. For example, imagine a character in a computer game, here the state of the system could be described in terms of the character's health and stat modifiers. We could also define the set of actions a character can perform and like in most good games we could introduce some uncertainty into the system (perhaps random enemy spawning). Then we could begin to investigate interesting objectives such as maximising the expected time a character stays alive in a game or maximize the expected score and translate these to strategies which lead to best outcomes.

Finite Horizon problems

Problems where the process terminates within a finite number of epochs are referred to as finite horizon problems. And it is useful to reinforce notions of a state space (which is the set of all states which are used to describe the system), an action space, which is the set of actions we can impose on the system, along with the outcome space which exist in the state space, it details the set of possible outcomes which are generally but not exclusively stochastic realisations.

We typically assume we can construct a transition matrix to describe the dynamic evolution of the system as well for mathematical convenience the state and action space are also assumed to be finite (although this is not necessarily the case).

If we have a finite horizon problem and satisfy these assumptions, we can apply Bellman's equation to solve MDPs. This is typically done by computing the value of all state-action pairs in a recursive manner, one possibility is using backwards value iteration solving Bellmans equation 1.0 for each possible state $s_t \in S$, maximising over all actions $a \in A_S$ and all outcomes $s' \in S$.

$$V_t(s_t) = \max_{a \in \mathcal{A}_S} (C_t(s_t, a) + \gamma \sum_{s' \in S} P(s'|s_t, a) V_{t+1}(s')) \tag{1}$$

Threshold policies

It is important to consider the structure of an MDP and there are some problems that naturally lend themselves to threshold polices. For instance for a stock replacement problem, where if at anytime the stock falls below a threshold level of stock σ, then we want to order a suitable amount of stock to reach Σ units. A decision rule may be expressed as:

$$d_t(s) = \begin{cases} \Sigma - s & s < \sigma \\ \\ 0 & s \geq \sigma \end{cases}$$

Where s is our current level of stock.

It is fairly obvious that if there exists a threshold (σ,Σ) optimal policy then there can be significant computational advantages. Furthermore, this can be translated to a simple rule of thumb, which is to aim for a target stock Σ and try to keep a minimum fill $\Sigma - \sigma$. This translates well for basic implemention and managerial purposes.

Infinite Horizon problems

Decisions in MDPs affect the states that will be visited later down the track and in turn will influence the future expected rewards. So the optimal choice of actions needs to consider future downstream impacts, which can expand many time periods into the future.

Infinite horizon problems are posed to investigate limiting behaviour (steady-state problems without the time dimension) to obtain insights into the properties of problems and algorithms. Unlike finite horizon problems which can be solved exactly using backward recursion because there exists some finite terminal condition, in the infinite setting applying value iteration would lead to costs tending to infinity.

This is worked around by considering the discounting cost model (which shrinks/contracts) per time period cost as $T \to \infty$ or alternatively we can use the average reward approach which divides total reward/cost by the number of stages, so that costs do not diverge to infinity. So to solve these problems we need to look at formulating the problem as either a Discounted MDP or Average Reward MDP.

Discounted MDP

In various applications a popular assumption (especially in finance) is to favour direct rewards over future uncertain rewards and so a parameter $\gamma \in [0, 1)$ can be introduced as a discounting factor. Where γ has an important role in rate of convergence of value and policy iteration algorithms along with allowing solution methology for infinite horizon problems.

Average Reward MDP

Previously we have assumed the objective function maximizes the reward or contribution per time period or epoch. But in some particular applications we might rather be interested in maximising the average reward per time period. That is we can analyse behaviour by observing what happens when T time periods becomes very large (approaches ∞).

Linear Programming approach for MDP An alternative method to find the optimal value function is by solving a linear programming problem such as follows.

$$\min_{v} \sum_{s \in S} \alpha_s v(s)$$

$$st. \quad v(s) \geq C(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v(s') \quad \forall s, \forall a$$

The advantage of the LP method over value iteration methods is that it avoids the need for iterative learning with the geometric convergence exhibited by value iteration. [4]

But the issue is the LP will have $|S| \times |A|$ inequality contraints and a $|S|$-dimensional decision vector, which can be unwieldy due to the tendency for the state space to be very large.

Partially Observable Markov Decision Processes (POMDPs)

There are many applications where we are not able to observe (or measure) the state of a system precisely. In these cases we may be able to model the application as a POMDP, which can be used to model a wide variety of interesting problems. Examples include robot navigation problems and automated planning (e.g. Google's self-driving car project). These problems are referred to as partially observable Markov decision processes, because the underlying state can not be directly observable and so we need to consider a probability distribution over the set of possible states based on a set of observations and observation probabilities.

Here an optimal solution will specify the optimal action for each possible belief over the state space. Unfortunately this comes at a cost, as finite horizon POMDPs are found to be PSPACE-hard as discussed in [2]. So these problems are largely intractable for all but the smallest of cases.

Model Predictive Control

There exists a class of control algorithms known as Model predictive control (MPC) alternatively known as 'receding horizon control' that takes advantage of an explicitly defined process model to predict how a plant responds in the future. Terminology such as process model and plant are abundant in MPC because it was initially developed for controlling power plant and petroleum refineries processes. [3]

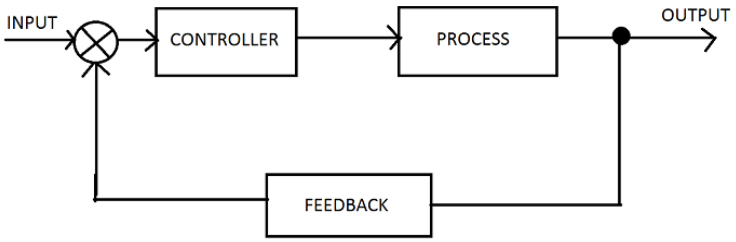

Figure 1: Generic block diagram of control system

The basic idea is that at each decision epoch, MPC algorithm computes a sequence of future 'manipulated variable adjustments' so to optimize future plant behaviour. Which is resolved at every subsequent control epoch. MPC is a computationally cheap algorithm that can deal with small unexpected disturbances although it is not quaranteed to find optimal policies.[3]

Concluding remarks

Control theory is born from a diverse range of fields which have contributed methods to the application of solving real-life problems. Although each field has different problems and different solution approaches, they all have a common theme of making sure a problem is well-defined and there exists some notion of an optimal sequence of actions to apply to the problem. To give some perspective of the scope of the field, we primarily looked at the dynamic programming side of control theory and even then we only scratched the surface of the topic.

$\sim$

The world is very much unpredictable, despite our best efforts we are yet unable to control a hurricane. But we can find some solace in the fact, while we can't control it, we can still take steps to limit the damage.

Bibliography

[1] Bellman, R. (1957). Dynamic Programming. Princeton University Press.

[2] Christos H. Papadimitiou and John N. Tsitsiklis. (1987). The complexity of Markov decision processes. Mathematics of Operations Reserch, 12(3): 441-450.

[3] S. Joe Qin , Thomas A. Badgwell (2003). A survey of industrial model predictive control technology. Control Engineering Practice, 11: 733-764.

[4] Powell, W. (2011). Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley Series in Probability and Statistics.

[5] Puterman, M. L. (1994). Markov Decision Processes. New York: John Wiley and Sons.