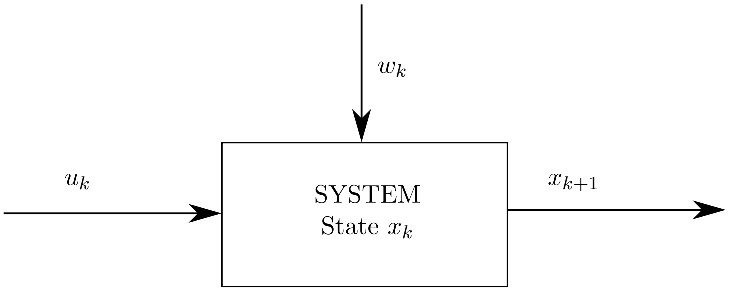
1. Introduction to Dynamic Programming

1.1 Problem Statement

Given a model of how a dynamic system evolves and a direct measurement of its state, apply a control input to the system so that a given cost is minimized.

Dynamics



$$x_{k+1} = f_k\left(x_k, u_k, w_k\right), \quad k = 0, 1, ..., N-1 \tag{1.1}$$

where

- k: discrete time index, or stage;
- N: given time horizon;
- $x_k \in \mathcal{S}_k$: system state vector at time k, can be measured at time k. $\mathcal{S}_k$ is the allowable set of states, which can be a function of time;
- $u_k \in \mathcal{U}_k(x_k)$: control input vector at time k. $\mathcal{U}_k(x_k)$ is the allowable set of control inputs, which can be a function of state and time;
- w_k: disturbance vector at time k, a random variable (see Appendix). It is assumed that w_k is conditionally independent with all prior variables x_l, u_l, w_l, $l < k$, given x_k and u_k. Furthermore, it is assumed that the conditional probability distribution of w_k given x_k and u_k is known;
- $f_k\left(\cdot, \cdot, \cdot\right)$: function capturing system evolution at time k.

Date compiled: September 19, 2023

Cost Function

We will consider the following *scalar-valued additive* cost function:

$$\underbrace{g_N(x_N)}_{\text{terminal cost}} + \underbrace{\sum_{k=0}^{N-1} \underbrace{g_k(x_k, u_k, w_k)}_{\text{stage cost}}}_{\text{accumulated cost}} \tag{1.2}$$

Example 1: Inventory Control

We are keeping an item stocked in a warehouse. If there is too little, we will run out of it and lose sales (not preferred). If there is too much, there will be more cost of storage and misuse of capital (not preferred). We will model this scenario as a discrete time system:

- $x_k \in \mathcal{S}_k = \mathbb{R}$: stock available in the warehouse at the beginning of the k^{th} time period.
- $u_k \in \mathcal{U}_k(x_k) = \mathbb{R}_{\geq 0}$: stock ordered and immediately delivered at the beginning of the k^{th} time period (supply).
- w_k: demand during the k^{th} time period, with some given probability distribution.
- *Dynamics*: $x_{k+1} = f_k(x_k, u_k, w_k) = x_k + u_k - w_k$. We assume that excess demand is back-logged, which corresponds to negative x_k.
- *Cost function*:

  $$R(x_N) + \sum_{k=0}^{N-1} r(x_k) + cu_k - pw_k$$

  where

  - pw_k: revenue
  - cu_k: cost of items;
  - $r(x_k)$: cost associated with too much stock or negative stock;
  - $R(x_N)$: terminal cost; cost associated with stock left at the end which we can't sell, or demand we can't meet;

△

Expected Cost

Let $X_1 := (x_1, \ldots, x_N)$, $U_0 := (u_0, \ldots, u_{N-1})$, and $W_0 := (w_0, \ldots, w_{N-1})$. Given x_0, variables X_1, U_0 and W_0 are all random variables due to the disturbances w_k and the dynamic coupling (1.1). For example, the state x_1 is a random variable with *probability density function* (PDF, see Appendix) defined through the system equations (1.1), the control input u_0, and the random variable w_0; x_2 is then a random variable as well, as it is a function of the random variables x_1, w_1, and so on. The control inputs can either be fixed and thus deterministic, or a function of the state and thus also random. The cost function (1.2) is thus a random variable; a convenient metric for optimization is taking its expected value to yield the expected cost of starting at an initial state x_0, that is,

$$\underset{(X_1,U_0,W_0|x_0)}{\mathrm{E}}\left[g_N(x_N)+\sum_{k=0}^{N-1} g_k(x_k,u_k,w_k)\right] \text{ subject to (1.1),}$$

where $\mathrm{E}\left[\cdot\right]$ is the *expectation operator*, see Appendix. Note: the underset on $\mathrm{E}\left[\cdot\right]$ is sometimes omitted for brevity. The expected cost can further be simplified as we will see in Section 1.2 depending on the employed control strategy.

1.2 Open Loop and Closed Loop Control

There are two different control methodologies: open loop, where all the control inputs are determined at once at time 0, and closed loop, where the control inputs are determined in a "just-in-time fashion", depending on the measured state x_k at time k.

1.2.1 Open Loop Control

Given an initial state x_0 and a set of control inputs $\bar{U}_0 := (\bar{u}_0,\dots,\bar{u}_{N-1})$ that is fixed (we use the bar to emphasize that the control inputs are fixed), the cost (1.2) becomes

$$g_N(x_N)+\sum_{k=0}^{N-1} g_k(x_k,\bar{u}_k,w_k)\,, \tag{1.3}$$

and is thus a function of the random variables x_k and w_k. We therefore wish to optimize the *expected open loop cost*

$$\underset{(X_1,W_0|x_0)}{\mathrm{E}}\left[g_N(x_N)+\sum_{k=0}^{N-1} g_k(x_k,\bar{u}_k,w_k)\right] \tag{1.4}$$

subject to the dynamics given by (1.1) (with $u_k=\bar{u}_k$).
Summarizing, the open loop control problem is thus the following: at time $k=0$, given x_0, find $\bar{U}_0$ that minimizes the expected open loop cost (1.4). This is called *open loop* control because measurements of the state are not used to calculate the control inputs.

1.2.2 Closed Loop Control

Let $\mu_k(\cdot)$ map state x_k to control input u_k:

$$u_k=\mu_k(x_k)\,,\;\; u_k\in\mathcal{U}_k(x_k)\;\forall x_k\in\mathcal{S}_k,\; k=0,\dots,N-1 \tag{1.5}$$

and define

$$\pi := \left(\mu_0(\cdot)\,,\mu_1(\cdot)\,,\dots,\mu_{N-1}(\cdot)\right),$$

where π is called an *admissible policy.* Given an initial state x_0, the states $x_1,\dots,x_N$, the control inputs $u_1,\dots u_{N-1}$ and the disturbances $w_0,\dots,w_{N-1}$ are random variables with PDFs defined through the system equations (1.1) and the state feedback equations (1.5).
The cost (1.2) therefore becomes

$$g_N(x_N)+\sum_{k=0}^{N-1} g_k(x_k,\mu_k(x_k),w_k)\,, \tag{1.6}$$

a function of the random variables $x_1, \ldots, x_N$ and $w_0, \ldots, w_{N-1}$. As a result, we define, for any $\mathrm{x} \in \mathcal{S}_0$, the *expected closed loop cost* associated with an admissible policy π to be

$$J_\pi(\mathrm{x}) := \underset{(X_1, W_0 | x_0 = \mathrm{x})}{\mathrm{E}} \left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right] \text{ subject to (1.1), (1.5).}$$

Let Π denote the set of all admissible policies. Then π^* is called an *optimal policy* if

$$J_{\pi^*}(\mathrm{x}) \leq J_\pi(\mathrm{x}) \qquad \forall \pi \in \Pi, \forall \mathrm{x} \in \mathcal{S}_0.$$

The *optimal cost* is defined as $J^*(\mathrm{x}) := J_{\pi^*}(\mathrm{x})$. $J^*(\cdot)$ is thus a function that maps initial states to optimal costs. The closed loop control problem aims at finding π^*.

Example 1: Inventory Control
For the inventory control problem described in Example 1, an intuitive example of an admissible policy is

$$\mu_k(x_k) = \begin{cases} s_k - x_k, & \text{if } x_k < s_k \\ 0, & \text{otherwise} \end{cases} \qquad k = 0, 1, \ldots, N-1$$

where s_k is some predefined, potentially time-varying threshold. △

1.2.3 Performance

It is clear that open loop control can never give better performance than closed loop control: a special case of closed loop control is to simply disregard state information, and thus open loop control is a special case of closed loop control. In the absence of disturbances w_k, the two give *theoretically* the same performance. In particular, in the absence of disturbances, $x_1, \ldots, x_N$ can be calculated from x_0 and $u_0, \ldots, u_{N-1}$, and there is thus no need to measure the state, and the optimal sequence of control inputs can be determined as soon as x_0 is known.
In practice, even when there are no disturbances, closed loop control will give better performance than open loop control for the following reasons:

- x_0 is often not known precisely and may also be random;
- the $f_k(\cdot, \cdot, \cdot)$ are often not known precisely;
- models are often only approximations of reality.

However, when a system is well-behaved and a good model for it exists, open loop control is a viable strategy, especially for short time horizons.

1.2.4 Computation

In terms of computation, open loop control is typically much less demanding than closed loop control. Consider, for example, a system with N_x distinct states and N_u distinct control inputs at every k^{th} time period. There are a total of N_u^N different open loop strategies and $N_u (N_u^{N_x})^{N-1} = N_u^{N_x(N-1)+1}$ different closed loop strategies. There are thus many more closed loop strategies than open loop ones. As an example, let's take $N_u = 10$, $N_x = 10$, and $N = 4$. The number of open loop strategies is 10^4. The number of closed loop strategies is 10^{31}, which is almost 10 orders of magnitude larger than the number of stars in the observable universe!

1.3 Discrete State and Finite State Problems

When state x_k takes on discrete values, or is finite in size, it is often more convenient to express the dynamics in terms of *transition probabilities*

$$P_{ij}(u,k) := \Pr\left(x_{k+1} = j \mid x_k = i, u_k = u\right) = p_{x_{k+1}|x_k,u_k}(j|i,u)$$

where $p_{x_{k+1}|x_k,u_k}(\cdot\,|\,\cdot,\,\cdot)$ denotes the PDF of x_{k+1} given x_k and u_k.
Given the transition probabilities, a system can be described equivalently with the dynamics

$$x_{k+1} = w_k$$

where w_k has the following probability distribution:

$$p_{w_k|x_k,u_k}(j|i,u) = P_{ij}(u,k).$$

Conversely, given a system with the dynamics $x_{k+1} = f_k\left(x_k, u_k, w_k\right)$ and $p_{w_k|x_k,u_k}(\cdot\,|\,\cdot,\,\cdot)$, we can provide an equivalent transition probability description, where the transition probabilities are

$$P_{ij}(u,k) = \sum_{\{\bar{w}_k | f_k(i,u,\bar{w}_k)=j\}} p_{w_k|x_k,u_k}(\bar{w}_k|i,u)$$

that is, $P_{ij}(u,k)$ is equal to the sum over the probabilities of all possible disturbances $\bar{w}_k$ that get us to state j from state i using control u at time k.

Example 1: Optimizing chess match strategy
Consider a two-game chess match with an opponent. Our objective is to come up with a strategy that maximizes the chance of winning the match. Each game can have one of two outcomes: 1) Win/Lose: 1 point for the winner, 0 for the loser; 2) Tie: 0.5 points for each player. In addition, if at the end of two games the score is equal, the players keep on playing new games until one wins, and thereby wins the match (also known as sudden death).
There are two possible playing styles for our player: timid and bold. When playing timid, our player ties with probability p_d and loses with probability $(1-p_d)$. When playing bold, our player wins with probability p_w and loses with probability $(1-p_w)$. We also assume that $p_d > p_w$, a necessary condition for this problem to make sense.
We will model this as a finite state problem:

- The *state* x_k is a two dimensional vector with the score of each player after the k^{th} game, where the first entry denotes the score of our player and the second the score of the opponent.

- The *control inputs* u_k are the two playing styles: timid and bold.

- The *disturbance* w_k is the score of the next game x_{k+1}.

- *Dynamics:* since it doesn't make sense to play timid if the game goes into sudden death, the problem is a two-stage finite state problem. We can construct a *Transition Probability Graph*, which can then be used to deduce $P_{ij}(u,k)$ to express the dynamics. Fig. 1.1 and 1.2 show all possible outcomes after the first and second game, respectively.

- *Cost*: we want to maximize the probability of winning, P_{win}. Therefore, the cost is $-P_{\text{win}}$, which is to be minimized. This is equivalent to the standard form

$$g_2(x_2) + \sum_{k=0}^{1} g_k(x_k, u_k, w_k)$$

where

$$g_k(x_k, u_k, w_k) = 0, \quad \forall k \in \{0, 1\}$$

$$g_2(x_2) = \begin{cases} -1 & \text{if } x_2 = (\frac{3}{2}, \frac{1}{2}) \text{ or } (2, 0), \\ -p_w & \text{if } x_2 = (1, 1), \\ 0 & \text{if } x_2 = (\frac{1}{2}, \frac{3}{2}) \text{ or } (0, 2). \end{cases}$$

Transition Probability Graph:

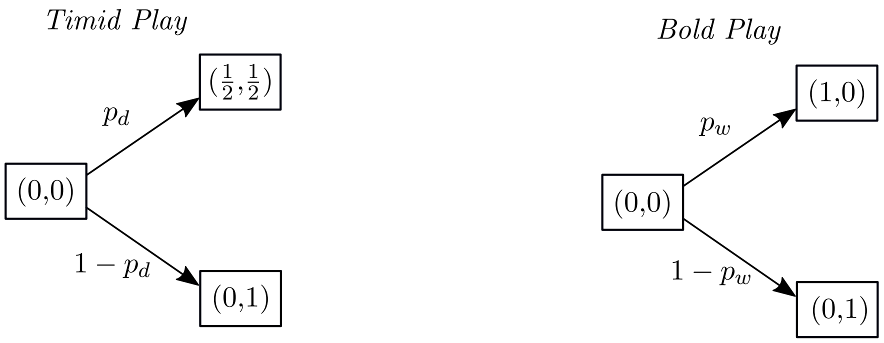
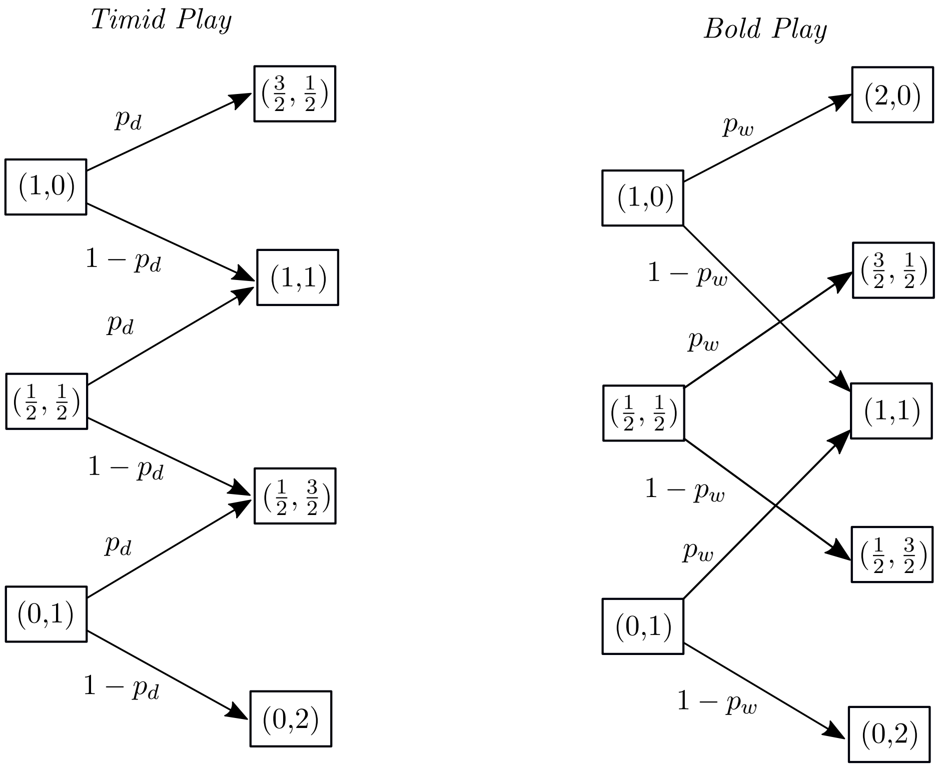


Figure 1.1: First game



Timid Play
$(\frac{3}{2},\frac{1}{2})$
p_d
(1,0)
$1 - p_d$
(1,1)
p_d
$(\frac{1}{2},\frac{1}{2})$
$1 - p_d$
$(\frac{1}{2},\frac{3}{2})$
p_d
(0,1)
$1 - p_d$
(0,2)
Bold Play
(2,0)
p_w
(1,0)
$1 - p_w$
$(\frac{3}{2},\frac{1}{2})$
p_w
(1,1)
$(\frac{1}{2},\frac{1}{2})$
$1 - p_w$
p_w
$(\frac{1}{2},\frac{3}{2})$
(0,1)
$1 - p_w$
(0,2)

Figure 1.2: Second game

Now let's look at the expected cost under open loop and closed loop strategies:

Open Loop Strategy: There are 4 possibilities:

1) Play timid in both games: $P_{\text{win}} = p_d^2 p_w$

2) Play bold in both games: $P_{\text{win}} = p_w^2 + p_w(1-p_w)p_w + (1-p_w)p_w p_w = p_w^2(3-2p_w)$

3) Play bold first, and timid in the second game: $P_{\text{win}} = p_w p_d + p_w(1-p_d)p_w$

4) Play timid first, and bold in the second game: $P_{\text{win}} = p_d p_w + (1-p_d)p_w^2$

Since $p_d^2 p_w \leq p_d p_w \leq p_d p_w + (1-p_d)p_w^2$, clearly 1) is not the optimal open loop strategy. The best achievable winning probability P^*_{win} is:

$$\begin{aligned} P^*_{\text{win}} &= \max\{\overbrace{p_w^2(3-2p_w)}^{2)},\ \overbrace{p_d p_w + (1-p_d)p_w^2}^{3)\text{ or }4)}\} \\ &= p_w^2 + \max\{2(p_w^2 - p_w^3),\ p_d p_w - p_d p_w^2\} \\ &= p_w^2 + \max\{2p_w(1-p_w)p_w,\ p_w(1-p_w)p_d\} \\ &= p_w^2 + p_w(1-p_w)\max\{2p_w,\ p_d\} \end{aligned}$$

If $p_d > 2p_w$, then 3) and 4) are the best open loop strategies, otherwise 2) is the best open loop strategy.

- For $p_w = 0.45$ and $p_d = 0.9$, $P^*_{\text{win}} = 0.43$.
- For $p_w = 0.5$ and $p_d = 1.0$, $P^*_{\text{win}} = 0.5$.

It can also be shown that, in the open loop case, if $p_w \leq 0.5$ then $P^*_{\text{win}} \leq 0.5$.

Closed Loop Strategy: There are 8 admissible policies. Let's consider one possible policy: play timid if and only if the player is winning (in Lecture 3 we will show that this strategy is indeed the optimal policy). Fig. 1.3 shows the corresponding transition probabilities under this specific policy. Then, the associated probability of winning P_{win} is

$$p_d p_w + p_w((1-p_d)p_w + p_w(1-p_w)) = p_w^2(2-p_w) + p_w(1-p_w)p_d$$

- For $p_w = 0.45$ and $p_d = 0.9$, $P_{\text{win}} = 0.54$
- For $p_w = 0.5$ and $p_d = 1.0$, $P_{\text{win}} = 0.625$

Note that in the closed loop case we can achieve a winning probability larger than 0.5 even when p_w is less than 0.5.

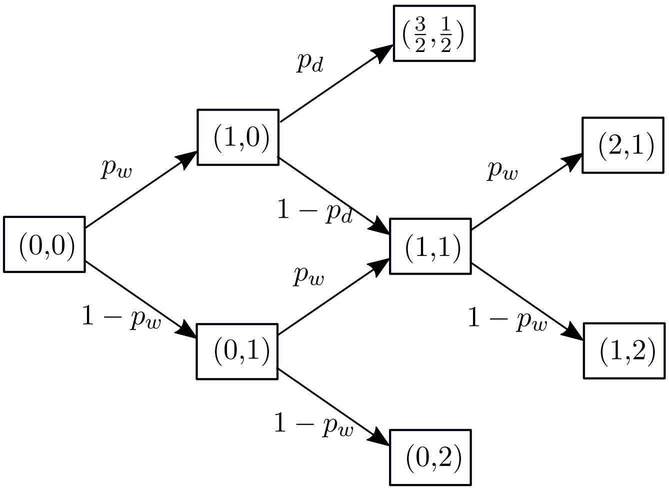


Figure 1.3: Closed loop strategy

△