Dynamic Programming Algorithm

Problem Definition

Underlying Discrete Time System

Dynamic programming requires a discrete time system model

$$x_{k+1} = f_k(x_k, u_k, w_k), \qquad k = 0, 1, \dots, N-1$$

$x_k \in S_k$: state $\quad u_k \in U_k$: input $\quad w_k \in D_k$: disturbance

- $u_k \in U(x_K) \subset C_k$, can be a function of current state.
- $w_k \sim \mathbb{P}_R(\cdot \mid x_k, u_k)$, noise can depend on state and input.

Additive Cost Function

Optimality has to be defined with respect to a certain cost

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

where g_k is a given, nonlinear function. Because *the disturbance is assumed to be random*, typically the expected value of the cost is considered

$$\mathop{\mathbb{E}}_{w_k}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right].$$

Transition Probability

When the state x_k is finite or discrete, transition probabilities are often convenient to describe the dynamics of a system

$$p_{ij}(u,k) := \mathbb{P}(x_{k+1} = j \mid x_k = i, u_k = u).$$

This is equivalent to $x_{k+1} = w_k$ with the following probability distribution for w_k:

$$\mathbb{P}(w_k = j \mid x_k = i, u_k = u).$$

Control Laws

Consider the control law

$$\Pi \ni \pi = \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$$

where μ_k maps state x_k to control u_k such that $\mu_k(x_k) \subset U(x_k), \forall x_k \in S_k$. Π is the set of all admissible policies.

- Given $\pi \in \Pi$, the expected cost of starting at state x_0 is

$$J_\pi(x_0) := \mathop{\mathbb{E}}_{w_k}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right].$$

- Optimal policy: $J_{\pi*}(x_0) \leq J_\pi(x_0) \forall \pi \in \Pi$.
- Optimal cost: $J^*(x_0) := J_{\pi*}(x_0)$

The DP Algorithm

For every initial state x_0, the optimal cost $J^*(x_0)$ of the basic problem is equal to $J_0(x_0)$, given by the last step of the following recursive algorithm, which proceeds backwards in time from $N-1$ to 0:

$$J_N(x_N) = g_N(x_N), \qquad k = 0, 1, \dots, N-1$$
$$J_k(x_k) = \min_{u_k} \mathop{\mathbb{E}}_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right]$$

where the expectation is taken with respect to $\mathbb{P}_{w_k}(\cdot \mid x_k, u_k)$. If $u_k^* = \mu_k^*(x_k)$ minimizes the recursion equation for all x_k and k, the policy $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$ is optimal. *For each recursion step, we have to perform the optimization over all possible states $x_k \in S_k$.*

Principle of Optimality

Assuming the policy $\pi^* = \{\mu_0^*, \dots \mu_{N-1}^*\}$ is optimal, the policy $\{\mu_i^*, \dots, \mu_{N-1}^*\}$ is optimal for a subproblem where $i > 0$.

Non-Standard Problems

Time Lags

Assume the system is of the form

$$x_{k+1} = f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k), \qquad k = 1, 2, \dots, N-1$$
$$x_1 = f_0(x_0, u_0, w_0).$$

Define the new state $\tilde{x}_k$ as

$$\tilde{x}_k = (\tilde{x}_{k,1} \quad \tilde{x}_{k,2} \quad \tilde{x}_{k,3})^\top := (x_k \quad x_{k-1} \quad u_{k-1})^\top.$$

This allows for the following notation of the new system

$$\tilde{x}_{k+1} = \begin{pmatrix} f_k(x_k, x_{k-1}, u_k, u_{k-1}, w_k) \\ x_k \\ u_k \end{pmatrix} = \begin{pmatrix} f_k(\tilde{x}_{k,1}, \tilde{x}_{k,2}, u_k, \tilde{x}_{k,3}, w_k) \\ \tilde{x}_{k,1} \\ u_k \end{pmatrix} = \tilde{f}_k(\tilde{x}_k, u_k, w_k).$$

Correlated Disturbances (Colored Noise)

If disturbances w_k are not independent, but can be modeled as the output of a system of independent disturbances ξ_k, i.e.,

$$w_k = C_k y_{k+1} \qquad y_{k+1} = A_k y_k + \xi_k \qquad k = 0, 1, \dots, N-1$$

a new state $\tilde{x}_k$ can be defined to return to the basic problem:

$$\tilde{x}_k := (x_k \quad y_k)^\top$$
$$\tilde{x}_{k+1} = \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \\ A_k y_k + \xi_k \end{pmatrix}.$$

In general y_k cannot be measured and must be estimated.

Forecasts

Assume that at the beginning of each time step k, it is known that the next disturbance w_{k+1} will be selected according to a particular probability distribution out of a given set of distributions $\{Q_1, \dots, Q_m\}$; i.e., if the forecast is i, $w_k \sim Q_i$. The a priori probability of the forecast being i is denoted by p_i. Model the forecast as

$$y_{k+1} = \xi_k$$

where ξ_k is a random variable, taking value i with probability p_i. Therefore, w_k has probability distribution Q_{y_k}. Define

$$\tilde{x}_k := (x_k \quad y_k)^\top$$

to arrive at the basic problem formulation

$$\tilde{x}_{k+1} = \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} f_k(x_k, u_k, w_k) \\ \xi_k \end{pmatrix}$$

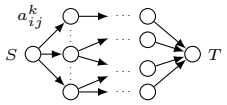
Note that the new disturbance $\tilde{w}_k = (w_k, \xi_k)$ depends on the current state, which is allowed, but not on prior disturbances. The DPA takes the following form:

$$J_N(x_N, y_N) = g_N(x_N), \qquad k = 0, 1, \dots, N-1$$
$$J_k(x_k, y_k) = \min_{u_k} \mathop{\mathbb{E}}_{w_k} \Big\{ g_k(x_k, u_k, w_k) + \sum_{i=1}^{m} p_i J_{k+1}(f_k(x_k, u_k, w_k), i) \,\Big|\, y_k \Big\}.$$

Here, the conditional expectation simply means $w_k \sim Q_{y_k}$.

Deterministic, Finite State Systems

Consider basic problems where $x_k \in S_k$, with $|S_k| < \infty$ and $w_k = 0 \; \forall k$. Assume that there is only one way to go from state $i \in S_k$ to $j \in S_{k+1}$ (if there is more, pick the one with lowest cost at stage k).



Let a_{ij}^k be the cost to go from state $i \in S_k$ to state $j \in S_{k+1}$ at time k (equal to ∞ if there's no way from i to j), particularly, a_{iT}^N is the terminal cost of state $i \in S_T$ at time N:

$$a_{ij}^k = g_k(i, u_k^{ij}), \quad \text{where } j = f_k(i, u_k^{ij})$$
$$a_{iT}^N = g_N(i).$$

DP Algorithm for Shortest Path Problems

Define $J_k(i)$ as the optimal cost of getting from i to T in $N-k$ moves. The DP Algorithm for shortest path problems is

$$J_N(i) = a_{iT}^N \quad i \in S_N \quad \text{(Initialization)}$$
$$J_k(i) = \min_{j \in S_{k+1}} \left(a_{ij}^k + J_{k+1}(j) \right) \quad i \in S_k, \; k = 0, \dots, N-1$$

The optimal cost is $\tilde{J}_0(S)$ and is equal to the length of the shortest path from S to T. For N nodes ($S = 1, T = N$):

$i \backslash k$	1	2	$\cdots$	$N-1$	N
S	$J_1(1)$	$J_2(1)$	$\cdots$	$J_{N-1}(1)$	$J_N(1)$
2	$J_1(2)$	$J_2(2)$	$\cdots$	$J_{N-1}(2)$	$J_N(2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$N-1$	$J_1(N-1)$	$J_2(N-1)$	$\cdots$	$J_{N-1}(N-1)$	$J_N(N-1)$
T	0	0	$\cdots$	0	0

$i \backslash k$	1	2	$\cdots$	$N-1$	N
S	$\mu_1(1)$	$\mu_2(1)$	$\cdots$	$\mu_{N-1}(1)$	$\mu_N(1)$
2	$\mu_1(2)$	$\mu_2(2)$	$\cdots$	$\mu_{N-1}(2)$	$\mu_N(2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$N-1$	$\mu_1(N-1)$	$\mu_2(N-1)$	$\cdots$	$\mu_{N-1}(N-1)$	$\mu_N(N-1)$
T	–	–	$\cdots$	–	–

Note: If node T can not be reached from node i in $\leq k$ steps, $J_k(i) = \infty$.

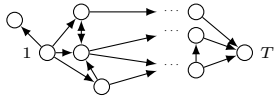
Forward DP Algorithm for Shortest Path Problems

Since the problem is symmetric, the shortest path from S to T is the same as from T to S, motivating the following algorithm:

$$\tilde{J}_N(j) = a_{Sj}^0 \qquad j \in S_1$$
$$\tilde{J}_k(j) = \min_{i \in S_{N-k}} \left(a_{ij}^{N-k} + \tilde{J}_{k+1}(i) \right) \qquad j \in S_{N-k+1}, \; k = 1, \dots, N$$

The optimal cost is $\tilde{J}_0(T)$ and is equal to the length of the shortest path from S to T.

Converting Shortest Path to DP



- Let $\{1, 2, \dots, N, T\}$ be a set of nodes of a graph. Let a_{ij} be the cost of moving from node i to node j, with $a_{ij} \to \infty$ if there is no path from i to j.
- Assume all cycles have non-negative cost. With this assumption, the length of an optimal path is bounded by N moves.
- Formulate the problem such that exactly N moves are required, but allow degenerate moves $a_{ii} = 0$.
- Terminate procedure if $J_k(i) = J_{k+1}(i) \; \forall i$.

Consider the following interpretation:

$J_k(i)$ = Optimal cost of getting from i to T in $N-k$ moves.

The DP algorithm then takes the form

$$J_{N-1}(i) = a_{iT}$$
$$J_k(i) = \min_j (a_{ij} + J_{k+1}(j)) \qquad k = 0, \dots, N-2.$$

Viterbi Algorithm

Objective

Suppose a Markov Chain with state transition probabilities p_{ij}

$$p_{ij} = P(x_{k+1} = j \mid x_k = i) \quad 1 \leq i, j \leq M,$$

and $p(x_0)$ as the probability for the starting state is given. Assume that the states can only be observed *indirectly*, i.e., via measurement:

$$r(z; i, j) = P(\text{meas} = z \mid x_k = i, x_{k+1} = j) \quad \forall k,$$

where P is the likelihood function. Given a set of measurements $Z_N = \{z_1, \dots, z_N\}$, the goal is to construct estimates for the states $\hat{X}_N = \{\hat{x}_0, \dots, \hat{x}_N\}$ that maximizes the probability $P(X_N \mid Z_N)$ over all $X_N = \{x_0, \dots, x_N\}$.

Solution

Recall $\mathbb{P}(X_N, Z_N) = \mathbb{P}(X_N \mid Z_N)\mathbb{P}(Z_N)$. Therefore, for a given Z_N, maximizing $\mathbb{P}(X_N, Z_N)$ over X_N gives the same result as maximizing $\mathbb{P}(X_N \mid Z_N)$ over X_N. The joint distribution can be rewritten as

$$\begin{aligned} \mathbb{P}(X_N, Z_N) &= \mathbb{P}(x_0, \dots, x_N, z_1, \dots, z_N) \\ &= \mathbb{P}(x_2, \dots, x_N, z_2, \dots, z_N \mid x_0, x_1, z_1) \\ &\quad \cdot \mathbb{P}(z_1 \mid x_0, x_1)\, \mathbb{P}(x_1 \mid x_0)\, \mathbb{P}(x_0) \\ &= \mathbb{P}(x_2, \dots, x_N, z_2, \dots, z_N \mid x_0, x_1, z_1) \\ &\quad \cdot r(z_1; x_0, x_1) p_{x_0 x_1} \mathbb{P}(x_0) \end{aligned}$$

One can keep going in this fashion to ultimately get

$$\mathbb{P}(X_N, Z_N) = \mathbb{P}(x_0) \prod_{k=1}^{N} p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k).$$

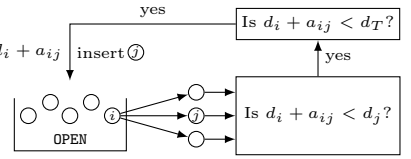
Assuming that all quantities are > 0, maximizing the joint distribution $\mathbb{P}(X_N, Z_N)$ can be rewritten as

$$\log\left(\mathbb{P}(X_N, Z_N)\right) =$$
$$\min_{X_N} \left(-\log(\mathbb{P}(x_0)) + \sum_{k=1}^{N} -\log\left(p_{x_{k-1}x_k} r(z_k; x_{k-1}x_k) \right) \right),$$

since the logarithm is a strict monotonically increasing function.

Label Correcting Methods

Assume $a_{ij} > 0$ and let d_i be the shortest path from i to S so far



0. Place node S in OPEN, set $d_s = 0, d_j = \infty \; \forall j$.
1. Remove a node i from OPEN, and execute step 2 for each child j of i.
2. If $d_i + a_{ij} < min(d_j, d_T)$, set $d_j = d_i + a_{ij}$, set i to be the parent of j. If $j \neq T$, place j in OPEN, replace if it's already there with higher d_j.
3. If OPEN is empty, you're done, otherwise go back to step 1.

Strategies for Removing Items from Open Bin

Different ways of removing items from the open bin result in different, well known algorithms:

- **Depth-First Search** Last in, first out. Finds feasible path quickly, good for limited memory.
- **Best-First Search** Also called Dijkstra's method. Remove best label (lowest d_i). Remove step is more expensive but can give good performance.
- **Breath-First Search** First in, first out. Bellman-Ford.

These algorithms can also be combined.

A^*–Algorithm

In step 2, the node j is only placed in the open bin if it satisfies $d_i + a_{ij} + h_j < d_T$, where h_j is a positive underestimate of the shortest distance from j to T.

Multi-Objective Problems

Pareto-Optimality

A vector $x = (x_1, \dots, x_M) \in S$ is non-inferior (pareto-optimal) if there are no other $y \in S$ so that $y_l \leq x_l, l = 1, \dots, M$, with strict inequality for one of the dimensions l, i.e., no vector y can perform better in any one dimension without being worse in at least one other dimension.

Extended Principle of Optimality

If $\{u_k, \dots, u_{N-1}\}$ is a non-inferior control sequence for the tail subproblem that starts at x_k, then $\{u_{k+1}, \dots, u_{N-1}\}$ is also non-inferior of the tail subproblem that starts at x_{k+1}.

Multi-Objective DPA Problem Setup

Given a problem with M cost functions $g^1(x), \dots, g^M(x)$, the vector $x \in X$ is a non-inferior solution of the multi-objective problem if the vector $(g^1(x), \dots, g^M(x))$ is a non-inferior vector of the set $\{(g^1(y), \dots, g^M(y) \mid y \in X\}$. The goal is to find all non-inferior control sequences.

$$x_{k+1} = f_k(x_k, u_k)$$

$$J^l_\pi(x_0) = g^l_N + \sum_{k=0}^{N-1} g^l_k(x_k, u_k) \qquad l = 1, \dots, M.$$

DPA for Multi-Objective Problems

Consider the following definitions:

- $F_k(x_k)$: The set of M-tuples of cost-to-go at x_k which are non-inferior. (Analogue to $J_k(x_k)$ for single-objective problems.)
- $F_N(x_N) := \{(g^l_N(x_N), \dots, g^M_N(x_N))\}$: The set of terminal costs. This set has only one element for each x_N.

The algorithm follows from the definitions above: Given $F_{k+1}(x_{k+1})\ \forall x_{k+1}$, generate for each state x_k the set of vectors $(g^1_k(x_k, u_k) + c^1, \dots, g^M_k(x_k, u_k) + c^M)$, such that the costs-to-go $(c^1, \dots, c^M) \in F_{k+1}(f_k(x_k, u_k))$ in order to find the new costs-to-go for all x_k. *$F_{k+1}(f_k(x_k, u_k))$ has to be computed for every state x_k, i.e. fix x_k and vary u_k to obtain result.* To obtain $F_k(x_k)$ extract all non-inferior elements of $(g^l_k(x_k, u_k) + c^1, \dots, g^M_k(x_k, u_k) + c^M)$.

Handling Constraints

Consider a cost function and $M-1$ constraints:

$$\text{cost:} \qquad g^1_N(x_N) + \sum_{k=0}^{N-1} g^1_k(x_k, u_k)$$

$$\text{constraints:} \qquad g^l_N(x_N) + \sum_{k=0}^{N-1} g^l_k(x_k, u_k) \le b^l,$$

where $l = 2, \dots, M$. The idea is to apply the DPA for multi-objective problems to compute the set of all non-inferior $F_0(x_0)$, subsequently throw away the elements that don't satisfy the constraints and then pick the one with the smallest cost.

A^*-Algorithm for Constraints

Let $\tilde{J}^l_k(x_k)$ be the optimal cost-to-arrive at x_k for $l = 2, \dots, M$, find them using DP (individually). Generate $\{g^l_k(x_k, u_k) + c^1, \dots g^M_k(x_k, u_k) + c^M\}$ such that

$$(c^1, \dots, c^M) \in F_{k+1}(f_k(x_k, u_k))$$

$$\text{and} \quad J^l_k(x_k) + g^l_k(x_k, u_k) + c^l \le b^l.$$

Continue as in the case above.

Infinite Horizon Problems

Consider the following time invariant system:

$$x_{k+1} = f_k(x_k, u_k, w_k), \qquad k = 0, 1, \dots, N-1$$

$$x_k \in S \qquad u_k \in U \qquad w_k \sim P(\cdot \mid x_k, u_k)$$

Note that neither S, U nor P are time dependent. Combine with a time invariant cost to go:

$$J_\pi(x_0) = \mathbb{E}\left[\sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), w_k)\right]$$

Note the lack of a terminal cost. The DP algorithm becomes

$$J_N(x_N) = 0$$

$$J_k(x_k) = \min_{u_k} \mathop{\mathbb{E}}_{w_k} \left[g(x_k, u_k, w_k) + J_{k+1}(f(x_k, u_k, w_k))\right]$$

For infinite horizons $N \to \infty$ we lose the notion of time.

Bellman Equation

The reasoning above gives rise to Bellman's equation:

$$J^*(x) = \min_u \mathop{\mathbb{E}}_w [g(x, u, w) + J^*(f(x, u, w))] \ \ \forall x \in S.$$

Note that one must solve for all $x \in S$ simultaneously.

Stochastic Shortest Path

The number of states is still finite. The transition probability from state i to j is given by $p_{ij}(u)$. We assume that there is a cost free termination state t with $p_{tt}(u) = 1$ and $g(t, u) = 0\ \forall u \in U$. The dynamics can be written as $x_{k+1} = w_k, P_R(w_k = j \mid x_k = i, u_k = u) = p_{ij}(u)$.

Value Iteration

Given any initial conditions $J_0(1), \dots, J_0(n)$, the sequence $J_k(i)$ generated by the DP iteration

$$J_{k+1}(i) = \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right\},$$

where $i = 1, \dots, n$, converges to the optimal cost $J^*(i)\ \forall i$.

Policy Iteration

Starting at a stationary policy μ^0 one can generate a sequence of new policies $\mu^1, \mu^2, \dots$ in the following way:

1. Given the policy μ^k, perform a policy evaluation step which computes $J_{\mu^k}(i), i = 1, \dots, n$, as the solution of the following linear system of n equations in the n unknowns $J(1), \dots, J(n)$:

$$J_{\mu^k}(i) = g(i, \mu^k(i)) + \sum_{j=1}^{n} p_{ij}(\mu^k(i)) J_{\mu^k}(j),$$

for $i = 1, \dots, n$.

2. Subsequently, perform a policy improvement step which computes a new policy μ^{k+1} as

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left\{ g(i, u) + \sum_{j=1}^{n} p_{ij}(u) J_{\mu^k}(j) \right\},$$

for $i = 1, \dots, n$. This process is repeated until $\mu^{k+1} = \mu^k$ for all i, in which case the algorithm terminates with the optimal policy μ^k.

Note that a mixture between Value and Policy iteration is also possible, since:

- Any number of values can be updated between policy updates;
- Any number of states can be updated at each value update;
- Any number of states can be updated at each policy update.

Linear Programming

It can be shown that the solution the the linear program

$$\max_J \sum_{i=1}^{n} J(i)$$

$$J(i) \le g(i, u) + \sum_{j=1}^{n} p_{ij}(u) J(j)\ \forall i,\ \forall u$$

yields the optimal costs $J^*(1), \dots, J^*(n)$ of the stochastic shortest path problem.

Discounted Problems

Given a policy π and a discount factor $\alpha \in (0, 1)$, the cost for infinite horizon $N \to \infty$ becomes

$$J_\pi(i) = \lim_{N \to \infty} \mathop{\mathbb{E}}_{x_k} \left[\sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)) \middle| x_0 = i \right].$$

Note that dynamics of the form $x_{k+1} = w_k, P_R(w_k = j \mid x_k = i, u_k = u) = p_{ij}(u)$ are still assumed. In this case, no termination state and no assumptions on the transition probabilities are required. Bellman's equation becomes:

$$J^*(i) = \min_{u \in U(i)} \left\{ g(i, u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right\} \ \forall i$$

Markov Decision Processes (MDPs)

Recall the cost of infinite horizon problems:

$$J_\pi(i) = \lim_{N \to \infty} \mathop{\mathbb{E}}_{x_k} \left[\sum_{k=0}^{N-1} g(x_k, \mu_k(x_k)) \middle| x_0 = i \right].$$

The stage cost is the expected value over all possible values x_k can take, given $x_0 = i$:

$$\mathop{\mathbb{E}}_{x_k} [g(x_k, \mu_k(x_k))].$$

In MDPs, the following stage cost is used:

$$\mathop{\mathbb{E}}_{x_k, x_{k+1}} [\bar{g}(x_k, x_{k+1})] = \mathop{\mathbb{E}}_{x_k} \left[\mathop{\mathbb{E}}_{x_{k+1}} [\bar{g}(x_k, x_{k+1}) \mid x_k] \right].$$

The equation holds since $P_k(x_{k+1} = j \mid x_k = i) = p_{ij}(\mu_k(x_k))$ only depends on x_k and not on any previous states. Therefore:

$$\mathop{\mathbb{E}}_{x_{k+1}} [\bar{g}(x_k, x_{k+1}) \mid x_k] = \sum_j \bar{g}(x_k, j) p_{ij}(\mu_k(x_k)),\ i = x_k,$$

and the conversion to a stochastic shortest path problem is

$$g(x_k, \mu(x_k)) = \sum_j \bar{g}(x_k, j) p_{ij}(\mu_k(x_k)),\ i = x_k.$$

Continuous Time Optimal Control

Consider the following system

$$\dot{x}(t) = f(x(t), u(t)) \qquad 0 \le t \le T$$

$$x(0) = x_0 \qquad \text{(no noise)},$$

where $x \in \mathbb{R}^n$ is the state, $t \in \mathbb{R}$ the time and $T \in \mathbb{R}$ the terminal time. The control is $u(t) \in U \subset \mathbb{R}^m$, where u is a constraint set. The following assumptions are made:

- A solution exists and is unique on the interval $0 \le t \le T$;
- f and $\partial f / \partial x$ are continuous with respect to x, i.e., f is continuously differentiable (Less stringent assumption: f is Lipschitz);
- f is continuous with respect to u; and
- $u(t)$ is piecewise continuous.

Additive Cost Funtion

Similar to the DP case, the cost function should be additive over time:

$$h(x(T)) + \int_0^T g(x(t), u(t))\, \mathrm{d}t.$$

Here, g and h have to be continuously differentiable with respect to x, and additionally, g has to be continuous with respect to u.

The Hamilton Jacobi Bellman Equation (HJB)

The HJB is the continuous time analogue to the DP algorithm:

$$0 = \min_{u \in U} \left\{ g(x, u) + \frac{\partial J^*(t, x)}{\partial t} + \left(\frac{\partial J^*(t, x)}{\partial x} \right)^\top f(x, u) \right\} \quad \forall x, \forall t$$

$$\text{Boundary Condition: } J^*(T, x) = h(x) \quad \forall x.$$

As in the DP occurrence, $J^*(t, x)$ can be interpreted as the optimal cost to go at time t and state x.

Pontryagin Minimum Principle

Let $H(x, u, p)$ be the Hamiltonian of the following form:

$$H(x, u, p) = g(x, u) + p(t)^\top f(x, u)$$

where $p(t)$ is called co–state, and defined as

$$p(t) := \frac{\partial J^*(t, x^*(t))}{\partial x} \qquad p_0(t) := \frac{\partial J^*(t, x^*(t))}{\partial t}.$$

Neccessary but not sufficient conditions on the solution are:

$$\dot{x}^*(t) = \frac{\partial}{\partial p} H(x^*(t), u^*(t), p(t)) \qquad x^*(0) = x_0$$

$$\dot{p}(t) = -\frac{\partial}{\partial x} H(x^*(t), u^*(t), p(t)) \qquad p(T) = \frac{\partial}{\partial x} h(x^*(T))$$

$$u^*(t) = \arg \min_{u \in U} \{H(x^*(t), u, p(t))\}$$

$$H(x^*(t), u^*(t), p(t)) = \text{constant} \qquad \forall t \in [0, T]$$

Consistent with the definition of the additive cost function, h is the terminal cost. Note that if $f(x, u)$ is linear, U is a convex set, and h and g are convex, the conditions are *neccessary and sufficient.*

Time Varying Systems

For time varying systems $f(x, u, t)$ and cost $g(x, u, t)$ the Hamiltonian is no longer constant along a trajectory.

Extensions to the Minimum Principle

Fixed Terminal State

Suppose that in addition to the initial state x_0, the terminal state $x(T)$ is also given. Since it is fixed, there is no need for a terminal cost. As the boundary condition for the co–state at terminal time $p(T)$ depends on the terminal cost, there is no meaningful constraint for it when the terminal state $x(T)$ is fixed. However, the problem is solvable as it has $2n$ ODEs and $2n$ boundary conditions:

$$\dot{x}(t) = f(x(t), u(t)) \qquad \dot{p}(t) = -\frac{\partial H(x(t), u(t), p(t))}{\partial x}$$

$$x(0) = 0 \qquad x(T) = x_T$$

Free Initial State with Cost

Suppose the initial state $x(0)$ is not fixed but subject to optimization, with associated cost $l(x(0))$. Consider a problem where the cost is

$$l(x(0)) + J(0, x(0))$$

at time $t = 0$. Since the second term is exactly $p(0)$ for optimal costs J^*, the condition on the optimal $p(0)$ becomes

$$p(0) = -\frac{\partial l(x(0))}{\partial x}.$$

Free Terminal Time

For problems where the terminal time is not fixed, one can show that the Hamiltonian not only has to be constant but equal to zero

$$H(x^*(t), u^*(t), p(t)) \equiv 0 \quad 0 \le t \le T.$$

Singular Problems

If the Hamiltonian $H(x^*(t), u^*(t), p(t))$ is independent of the input u for a nontrivial time interval, the minimum principle is insufficient to determine the optimal policy.

Linear Systems and Quadratic Cost

Infinite Horizon LTI System

$$x_{k+1} = Ax_k + Bu_k \qquad k = 0, 1, \dots$$

$$\text{cost}: \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \qquad R = R^T > 0, Q = Q^T \ge 0$$

(A, B) is stabilizable, (A, C) is detectable where C is any matrix that satisfies $C^T C = Q$. Then:

- $\exists$ unique solution
- optimal cost to go is $J(x) = x^T K x$
- optimal feedback strategy $u = Fx$
- resulting closed loop system is stable

We can derive K and F from the assumptions made above:

$$K = Q + A^T \left(K - KB\left(R + B^T K B\right)^{-1} B^T K\right) A, \quad K = K^T \ge 0$$

$$F = -\left(R + B^T K B\right)^{-1} B^T K A$$

Stabilizable and Observable

- **Stabilizable:** One can find a matrix F such that $A + BF$ is stable. $\rho(A + BF) < 1$, eigenvalues of $A + BF$ have magnitude < 1.
- **Observable:** Let Q be decomposed as $Q = C^T C$. (A, C) is detectable if $\exists L$ such that $A + LC$ is stable.