Linear Quadratic Optimal Control Topics

- Finite time LQR problem for time varying systems
  - Open loop solution via Lagrange multiplier
  - Closed loop solution
  - Dynamic programming (DP) principle
  - Cost-to-go function computed from DP
- Infinite time LQ problem for LTI systems
  - Convergence of $P(t \to -\infty, t_f)$
  - Closed loop stability
  - P_∞ as solution of ARE via Hamiltonian matrix
  - Selection of Q, R and S
- Some extensions
  - Discrete time LQ
  - Pole-placement within a pre-defined region
  - Frequency shaping

Linear Quadratic (LQR) Optimal Control - Motivation

- Pole-placement approach allows ones to choose where to place the poles
  - SI feedback gain unique
  - MI feedback gain non-unique (e.g. need Hautus-Keyman Lemma or eigenvector placement)
- Main issue: where should we place the poles???
- Should consider trade-off between performance, robustness and control effort.
- **LQ technique tries to do some trade-off without specifying desired poles locations**

Finite Time Linear Quadratic Optimal Regulator

$m-$ input, $u \in \Re^m$, $n-$state system with $x \in \Re^n$:

$$\dot{x} = A(t)x + B(t)u; \qquad x(0) = x_0. \tag{1}$$

Find open loop control $u(\tau)$, $\tau \in [t_0, t_f]$ such that the following objective function is minimized:

$$J(u, x_0, t_0) = \frac{1}{2}x^T(t_f)Sx(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left[x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\right] dt \tag{2}$$

- $Q(t) = Q^T(t)$ and S are symmetric positive semi-definite $n \times n$ matrices
- $R(t) = R^T(t)$ is a symmetric positive definite $m \times m$ matrix.

Notice that x_0, t_0, and t_f are fixed and given data.

The control goal generally is to keep $x(t)$ close to 0[1], especially, at the final time t_f, using little control effort u. To wit, notice in (2)

- $x^T(t)Q(t)x(t)$ penalizes the transient state deviation,
- $x(t_f)^T S x(t_f)$ penalizes the finite state
- $u^T(t)R(t)u(t)$ penalizes control effort.

Output regulation:
If $y = C(t)x$ is the output, we can define:

$$Q(t) = C^T(t)W(t)C(t)$$

where $W(t)$ is a symmetric, positive definite output weighting matrix.

[1] LQ can be modified for the trajectory tracking case

General Finite Time Optimal Control

Plant:

$$\dot{x} = f(x, u, t); \qquad x(t_0) = x_0 \text{ given.}$$

Time interval: $t \in [t_0, t_f]$.
Cost function to be minimized:

$$J(u(\cdot), x_0) = \phi(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt$$

First term = *final cost* and the second term = *running cost*.
Problem: Find $u(t)$, $t \in [t_0, t_f]$ such that $J(x_0, u(\cdot))$ is minimized, subject $x(t)$ satisfying the plant equation $x(t_0) = x_0$ given.

Solution - Hamilton-Jacobi-Bellman Equations

IDEA: convert constrained optimal control into unconstrained optimal control using Lagrange multiplier $\lambda(t) \in \Re^n$:

$$\bar{J}(u, \lambda(\cdot), x_0) = J(u(\cdot), x_0) + \int_{t_0}^{t_f} \lambda^T(t)[f(x, u, t) - \dot{x}]dt.$$

Note that $\frac{d}{dt}(\lambda^T(t)x(t)) = \dot{\lambda}^T(t)x(t) + \lambda^T(t)\dot{x}(t)$. So

$$\int_{t_0}^{t_f} (\lambda^T \dot{x})dt = \lambda^T(t_f)x(t_f) - \lambda^T(t_0)x(t_0) - \int_{t_0}^{t_f} (\dot{\lambda}^T x)dt.$$

Let us define the so called Hamiltonian function

$$H(x, u, t) := L(x, u, t) + \lambda^T(t)f(x, u, t).$$

Necessary condition for optimality

Variation of the modified cost $\delta\bar{J}$ with respect to all feasible variations $\delta x(t)$ and $\delta u(t)$ and $\delta\lambda(t)$ should vanish.
Using integration by parts: $\frac{d}{dt}\lambda x = \lambda x - \dot{\lambda}x$,

$$\bar{J} = \phi(x(t_f)) - \lambda^T(t_f)x(t_f) + \lambda^T(t_0)x(t_0) + \int_{t_0}^{t_f} H(x(t), u(t), t) + \dot{\lambda}(t)x(t)]\, dt$$

$$\delta\bar{J} = [\phi_x - \lambda^T]\delta x(t_f) + \lambda^T(t_0)\delta x(t_0) + \int_{t_0}^{t_f} [H_x + \dot{\lambda}^T]\delta x + H_u \delta u dt + \int_{t_0}^{t_f} \delta\lambda^T[f(x(t), u(t), t) - \dot{x}]dt$$

Since $x(t_0) = x_0$ is fixed, $\delta x(t_0) = 0$. Otherwise, other variations $\delta x(t)$, $\delta u(t)$ or $\delta\lambda(t)$ are all feasible.

Cont'd

Hence,

$$\dot{\lambda} = -H_x = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x} \quad (3)$$

$$\dot{x} = f(x, u, t) \quad (4)$$

$$H_u = -\frac{\partial L}{\partial u} - \lambda^T \frac{\partial f}{\partial u} = 0 \quad (5)$$

$$\lambda^T(t_f) = \frac{\partial \phi}{\partial x}(x(t_f)) \quad (6)$$

$$x(t_0) = x_0. \quad (7)$$

This is a set of $2n$ differential equations (in x and λ) with split boundary conditions at t_0 and t_f: $x(t_0) = x_0$ and $\lambda^T(t_f) = \phi_x(x(t_f))$.

Finite Time LQ Regulator Solution

Open loop formulation:

$$L(x,u,t) = \frac{1}{2}x^T(t)Q(t)x(t) + \frac{1}{2}u^T(t)R(t)u(t)$$

$$\phi(x(t_f)) = \frac{1}{2}x^T(t_f)Sx(t_f)$$

$$f(x,u,t) = A(t)x + B(t)u$$

Using the above in Eqs.(3)-(7), the optimal control is (see (5)):

$$u^o(t) = -R^{-1}B^T(t)\lambda(t)$$

where $\lambda(t)$ and $x(t)$ satisfy the Hamilton-Jacobi eqn (3)-(4):

$$\begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \underbrace{\begin{pmatrix} A(t) & -B(t)R^{-1}B^T(t) \\ -Q(t) & -A^T(t) \end{pmatrix}}_{\text{Hamiltonian Matrix - } H(t)} \begin{pmatrix} x \\ \lambda \end{pmatrix} \tag{8}$$

with boundary conditions given by (see (6)-(7)):

$$x(t_0) = x_0; \qquad \lambda(t_f) = Sx(t_f).$$

Open loop solution - Remarks

- Boundary conditions specified at initial time t_0 and final time t_f (two point boundary value problem). In general, these are difficult to solve requires iterative methods such as *shooting method*.
- Optimal control is **open loop**. It is computed by first computing $\lambda(t)$ for all $t \in [t_0, t_f]$ and then applying $u^o(t) = -R^{-1}B^T(t)\lambda(t)$.
- Open loop control is not robust to disturbances or uncertainties.

Consider $X_1(t) \in \Re^{n\times n}$ and $X_2(t) \in \Re^{n\times n}$ satisfying the Hamilton-Jacobi equation:

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} A(t) & -B(t)R^{-1}B^T(t) \\ -Q(t) & -A^T(t) \end{pmatrix}}_{\text{Hamiltonian Matrix - } H} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

with $X_1(t_f)$ non-singular (e.g. $X_1(t_f) = I_{n\times n}$) & $X_2(t_f) = SX_1(t_f)$.

Requires solving the $2n \times n$ differential equations backward in time.

Sweeping method

Claim: *Assuming that $X_1(t)$ is invertible for all $t \in [t_0, t_f]$. Then, we can express $x(t)$ and $\lambda(t)$ satisfying the Hamilton-Jacobi equation by:*

$$\begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} = \begin{pmatrix} X_1(t) \\ X_2(t) \end{pmatrix} v$$

for some constant $v \in \Re^n$. Moreover,

$$\lambda(t) = [X_2(t) X_1^{-1}(t)] x(t)$$

Proof: By direct substitution. With $v = X_1^{-1}(t_0) x_0$, that $x(t)$ and $\lambda(t)$ satisfy the HJB equation and boundary conditions.

⇒ Closed Loop Control

- This implies that optimal control can be expressed as **closed loop state-feedback**:

$$u^o(t) = -R^{-1}B^T(t)\lambda(t) = -R^{-1}B^T(t)P(t)x(t)$$

  where $P(t) := X_2(t)X_1^{-1}(t) \in \Re^{n\times n}$.

- Note: $P(t)$ still needs to be solved first (backwards in time).

ODE for $P(t) = X_2(t)X_1^{-1}(t)$

Differentiating $P(t) := X_2(t)X_1^{-1}(t)$ and using Hamilton-Jacobi equation (for $X_1(t)$ and $X_2(t)$), we find that $P(t)$ satisfies the continuous time Riccati differential equation (CTRDE):

$$\dot{P}(t) = -A^T(t)P(t) - P(t)A(t) + P(t)B(t)R^{-1}(t)B^T(t)P(t) - Q(t); \tag{9}$$

with boundary condition $P(t_f) = S$.

- $P(t)$ is symmetric and positive semi-definite.
- Symmetric because S and all terms in (9) are symmetric
- To show that $P(t)$ is positive semi-definite, we will relate $P(t)$ to minimum cost:

$$J(u^o, x_0, t_0) = \frac{1}{2}x_0^T P(t_0)x_0.$$

where $u^o(t)$ is the optimal control.

Form of $J(u^o, x_0, t_0)$

- From the optimal control, and closed loop system being linear ....

$$u^o(t) = -R^{-1}(t)B^T(t)P(t)x(t)$$
$$x(t) = \Phi(t, t_0)x_0$$

- The form of the minimum cost function Eq.(2) must be:

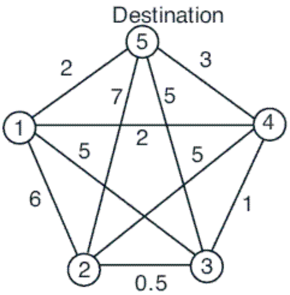
$$J^o(x_0, t_0) = J(u^o, x_0, t_0) = \frac{1}{2}x_0^T \bar{P}(t_0)x_0.$$

for some positive semi-definite matrix $\bar{P}(t_0)$.

- To show that $\bar{P}(t_0) = P(t_0)$, we need to understand the Dynamic Programming Principle.

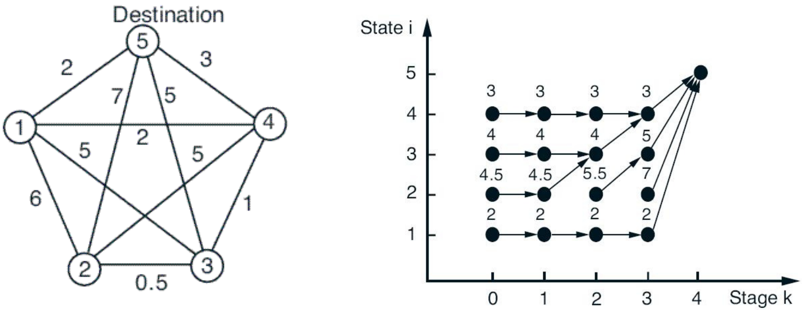
Dynamic Programming (DP) Principle

Consider a shortest path problem in which we need to traverse a network from state i_0 and to reach state 5 with minimal cost.



- Cost to traverse an arc from $i \to j$ is $a_{ij} > 0$.
- Cost to stay is $a_{ii} = 0$ for all i.
- Since there are only 4 non-destination states, state 5 can be reached in at most $N = 4$ steps.

DP example - cont'd

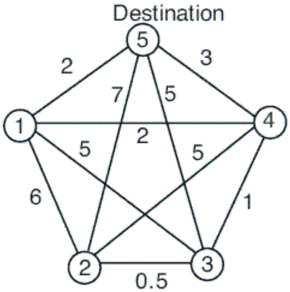


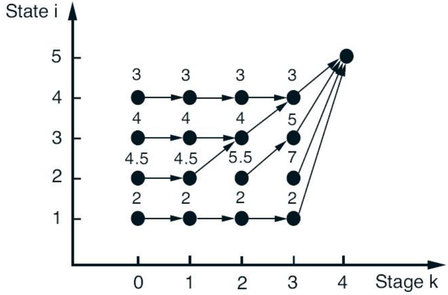
- Total cost is sum of the cost incurred, i.e. if the (non-optimal) control policy π is $2 \to 2 \to 3 \to 4 \to 5$, then

$$J(\pi) = a_{22} + a_{23} + a_{34} + a_{45}$$

- Goal is to find the policy that minimizes J.
- As an optimization problem the space of 4 step policy has a cardinality of $5^4 = 625$

DP algorithm:





- We start from the end stage ($N = 4$), i.e. you need to reach the state 5 in one step. Suppose that you are in state i, the cost to reach state 5 is

$$min\{a_{i5}\} = a_{i5}$$

- The optimal and only choice for next state, if currently at state i, is $u^*(i,N) = 5$. Optimal cost-to-go is $J^*(i,N) = a_{i5}$.

Node	1	2	3	4
u^*	5	5	5	5
$J^*(i,N)$	2	7	5	3

DP example - cont'd

- Consider the $N-1$st stage, and you are in state i. We can have the policy $\pi : i \to j \to 5$. Since the minimum cost to reach state 5 from state j is $J^*(j,N)$, the optimal control policy is:

$$\min_j (a_{ij} + J^*(j,N))$$

$$= \min\{a_{i1} + a_{15}, a_{i2} + a_{25}, \ldots, a_{i5} + a_{55}\}$$

For $i = 4$ (for instance),

j	1	2	3	4
a_{4j}	2	5	1	0
$a_{4j} + J^*(j,N)$	4	11	6	3

Thus, the j the optimizes this is: $j = 4$ (stay put) so that $u^*(4, N-1) = 4$ and $J^*(4, N-1) = 3$.

Doing this for each i, we have at stage $N-1$,

- Optimal policy:

$$u^*(i, N-1) = arg\min_j(a_{ij} + J^*(j, N))$$

- Optimal cost-to-go:

$$J^*(i, N-1) = \min_j(a_{ij} + J^*(j, N))$$

Node	1	2	3	4
$u^*(i, N-1)$	1	3	4	4
$J^*(i, N-1)$	2	5.5	4	3

DP example - Cont'd

- If we are at the $N-2$nd stage, and you are in state i,
  - Optimal policy:

$$u^*(i, N-2) = \arg\min_j (a_{ij} + J^*(j, N-1))$$

  - Optimal cost-to-go:

$$J^*(i, N-2) = \min_j (a_{ij} + J^*(j, N-1))$$

Node	1	2	3	4
$u^*(i, N-2)$	1	3	3	4
$J^*(i, N-2)$	2	4.5	4	3

Notice that from state 2, the 3 step policy

$$2 \to 3 \to 4 \to 5$$

has a lower cost of 4.5 than the 2 step policy $2 \to 3 \to 5$ with a cost of 5.5.

- Repeating the propagation procedure for the optimal policy and optimal cost-toabove until $N = 1$. Then the optimal policy is $u^*(i, 1)$ and the minimum cost is $J^*(i, 1)$.
- The optimal sequence starting at i_0 is:

$$
\begin{aligned}
i_0 &\rightarrow u^*(i_0, 1) \rightarrow u^*(u^*(i_0, 1), 2) \\
&\rightarrow u^*(u^*(u^*(i_0, 1), 2), 3) \rightarrow 5
\end{aligned}
$$

Remarks

- At each stage k, the optimal policy $u^*(i,k)$ is a state feedback policy. i.e. it determines what to do depending on the state that you are in.
- Policy and optimal cost-to-go are computed backwards in time (stage)
- At each stage, the optimization is done on the space of intermediate states, which has a cardinality of 5.
- The large optimization problem with cardinality of $5^4 = 624$ has been reduced to 4 simpler optimization problem with cardinality of 5 each ($4 \times 5 = 20$).

Dynamic Programming Principle

"The tail end of the optimal sequence is optimal for the tail problem".

- If the optimal 4 step sequence π_4 starting at i_0 is:

$$i_0 \to u^*(i_0, 1) \to u^*(u^*(i_0, 1), 2) \to u^*(u^*(u^*(i_0, 1), 2), 3) \to 5$$

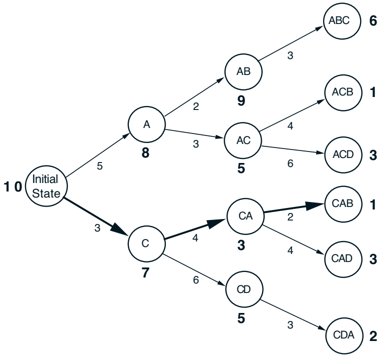
- then the sub-sequence π_2

$$u^*(i_0, 1) \to u^*(u^*(i_0, 1), 2) \to u^*(u^*(u^*(i_0, 1), 2), 3) \to 5$$

  is the optimal 3 step sequence starting at $u^*(i_0, 1)$.
- This is so because if $\bar{\pi}_3$ is another 3 step sequence starting at $u^*(i_0, 1)$ with a strictly lower cost than π_3, then the 4-step sequence $i_0 \to \bar{\pi}_3$ will also have a lower cost than $\pi_4 = i_0 \to \pi_3$ which is assumed to be optimal.

Another DP example

- Find optimal sequence of operations A, B, C, D
(A must precede B and C must precede D)



Bold numbers are the values of the cost-to-go function for the node (and stage).

System:

$$\dot{x} = f(x(t), u(t), t), \qquad x(t_0) = x_0,$$

Cost index:

$$J(u(\cdot), t_0) = \int_{t_0}^{t_f} L(x(t), u(t), t)dt + \phi(x(t_f)). \tag{10}$$

Continuous Time DP - cont'd

Suppose that $u^o(t)$, $t \in [t_0, t_f]$ *minimizes* (10) *subject to* $x(t_0) = x_0$ *and* $x^o(t)$ *is the associated state trajectory. Let the minimum cost achieved using* $u^o(t)$ *be:*

$$J^o(x_0, t_0) = argmin_{u(\tau), \tau \in [t_0, t_f]} J(u(\cdot), t_0)$$

Then, for any Δt *s.t.* $t_0 \leq t + \Delta t \leq t_f$*, the restriction of the control* $u^o(\tau)$ *to* $\tau \in [t + \Delta t, t_f]$ *minimizes*

$$J(u(\cdot), t_0 + \Delta t) = \int_{t_0 + \Delta t}^{t_f} L(x(t), u(t), t) dt + \phi(x(t_f)).$$

subject to initial condition $x(t_0 + \Delta t) = x^o(t_0 + \Delta t)$.
i.e. $u^o(\tau)$ *is optimal over the sub-interval.*

Typical application of DP

- Solve the optimal control problem for sub-interval $[t_1, t_f]$ with **arbitrary** initial states, $x(t_1) = x_1$.
- Let the control that is optimal be $u(t) = u^o(t, t_1, x_1)$ and let $J^o(x_1, t_1)$ be the optimal cost given initial state $x(t_1) = x_1$.
- Now consider $t_0 < t_1$. The optimal control $u^o(t, t_0, x_0)$ for the interval $[t_0, t_f]$ with initial states, $x(t_0) = x_0$ is given as follows.
- For $t_0 \leq t \leq t_1$, $u^o(t, t_0, x_0)$ is the $u(t)$ that minimizes:

$$\int_{t_0}^{t_1} L(x(t), u(t), t)dt + J^o(x(t_1), t_1)$$

subject to $\dot{x}(t) = f(x(t), u(t), t)$. Notice that $x(t_1)$ is unknown a-priori since it depends on $u(t)$.

- For $t_1 \leq t \leq t_f$, the optimal control

$$u^o(t, t_0, x_0) = u^o(t, t_1, x(t_1))$$

  where $x(t_1)$ is the state achieved at $t = t_1$ from the initial state x_0 using optimal control $u^o(t, t_0, x_0)$ over the interval $[t_0, t_1]$.
- This procedure can be repeated by taking the initially time further and further back.
- The optimal cost $J^o(x, t)$ is the **cost-to-go function** at time t.

Relating $P(t)$ to cost-to-go

- Let us apply DP to the LQ case (note: without the 1/2 for simplicity):

$$L(x,u,t) = \frac{1}{2}x^T Q(t)x + \frac{1}{2}u^T R(t)u$$
$$f(x,u,t) = A(t)x + Bu$$
$$J = \int_{t_0}^{t_f} L(x,u,t)dt + \phi(x(t_f)).$$

- At $t = t_f$, the cost-to-go function is simply:

$$J^o(x,t_f) = \frac{1}{2}x^T(t_f)Sx(t_f) = \frac{1}{2}x^T(t_f)\bar{P}(t_f)x(t_f)$$

- Hence, $\bar{P}(t_f) = S$.

Relating $P(t)$ to cost-to-go - 2

- Let $t_1 = t_f$ and consider $t = t_1 - \Delta t$ where Δt is infinitesimally small.
- The optimal control at t given the state $x(t)$ is minimize

$$min_{u(t)} L(x, u, t)\Delta t + J^o(x(t_1), t_1)$$

- Now, $x(t_1) = x(t) + f(x(t), u(t), t)\Delta t$. Thus, we minimize w.r.t. $u(t)$,

$$\frac{1}{2}\left[x(t)^T Q(t)x(t) + u^T(t)R(t)u(t)\right]\Delta t +$$
$$J^o\left(x(t) + [A(t)x(t) + B(t)u(t)]\Delta t, t_1\right)$$
$$\approx \frac{1}{2}\left[x(t)^T Q(t)x(t) + u^T(t)R(t)u(t)\right]\Delta t + \frac{1}{2}x(t)\bar{P}(t_1)x(t)$$
$$+\frac{1}{2}[x^T(t)A^T(t) + u^T(t)B^T(t)]\bar{P}(t_1)x(t)\Delta t$$
$$+\frac{1}{2}x^T(t)\bar{P}(t_1)[A(t)x(t) + B(t)u(t)]\Delta t$$

- Differentiating w.r.t. $u(t)$, we get the optimal control policy:

$$u^{oT} R(t) + x^T(t)\bar{P}(t_1)B(t) = 0$$
$$\Rightarrow u^o(t) = -R^{-1}(t)B^T(t)\bar{P}(t_1)x(t)$$

- The updated optimal cost-to-go function is:

$$\begin{aligned} 2 \cdot J^o(x(t), t) \approx & \left[x(t)^T Q(t)x(t) + u^{oT}(t)R(t)u^o(t)\right] \Delta t \\ & + x(t)\bar{P}(t_1)x(t) \\ & + [x^T(t)A^T(t) + u^{oT}(t)B^T(t)]\bar{P}(t_1)x(t)\Delta t \\ & + x^T(t)\bar{P}(t_1)[A(t)x(t) + B(t)u^o(t)]\Delta t \end{aligned}$$

- This shows that

$$
\begin{aligned}
& 2 \cdot J^o(x(t), t) \\
\approx\ & x^T(t)\bar{P}(t_1)x(t) + x^T(t)\Big[A^T(t)\bar{P}(t_1) + \bar{P}(t_1)A(t) \\
& -\bar{P}(t_1)B(t)R^{-1}(t)B^T(t)\bar{P}(t_1) + Q(t)\Big]x(t) \cdot \Delta t \\
=\ & x^T(t)\bar{P}(t)x(t)
\end{aligned}
$$

where

$$
\begin{aligned}
& -\left(\bar{P}(t_1) - \bar{P}(t)\right) \\
=\ & \Big[A^T(t)\bar{P}(t_1) + \bar{P}(t_1)A(t) \\
& -\bar{P}(t_1)B(t)R^{-1}(t)B^T(t)\bar{P}(t_1) + Q(t)\Big]\Delta t \qquad (11)
\end{aligned}
$$

- Thus, we have shown that at t,

$$J^o(x(t), t) = \frac{1}{2} x^T(t)\bar{P}(t)x.$$

- Let $t \to t_1$, $t - \Delta t \to t$ and repeat the process and we get the update recursion in Eq.(11).
- As $\Delta t \to 0$, we have Eq.(11) becomes:

$$\begin{aligned} -\dot{\bar{P}}(t) = A^T(t)\bar{P}(t) + \bar{P}(t)A(t) \\ - \bar{P}(t)B(t)R^{-1}(t)B^T(t)\bar{P}(t) + Q(t); \end{aligned}$$

  which is exactly the Riccati differential equation as before.
- Together with $\bar{P}(t_f) = P(t_f) = S$, this shows that $\bar{P}(t) = P(t)$.

- Note: Since

$$\begin{aligned} & x^T(t)P(t)x(t) \\ = & \int_t^{t_f} \left[x^T(\tau)Q(\tau)x(\tau) + u^T(\tau)R(\tau)u(\tau) \right] d\tau + x^T(t_f)Sx(t_f) \\ \geq & \ 0 \end{aligned}$$

for any $x(t)$, $P(t)$ is **positive semi-definite** for any $t \leq t_f$.

Finite time LQ Summary

- The finite time LQ regulator problem is solved by the control:
$$u^*(t) = -R^{-1}(t)B^T(t)P(t)x(t) \tag{12}$$
where $P(t) \in \Re^{n\times n}$ is the solution to the continuous time Riccati Differential Equation (CTRDE):
$$\begin{aligned}\dot{P}(t) = & -A^T(t)P(t) - P(t)A(t) \\ & + P(t)B(t)R^{-1}(t)B^T(t)P(t) - Q(t);\end{aligned}$$
with boundary condition $P(t_f) = S$.
- $P(t)$ is positive-semi definite
- The minimum cost achieved using the above control:
$$J^*(x_0, t_0) := min_{u(\cdot)} J(u, x_0) = \frac{1}{2} x_0^T P(t_0) x_0$$

Further Remarks - Finite Time LQ

- The control formulation works for time varying systems, e.g. nonlinear systems linearized about a trajectory.
- The optimal control law is in the form of a time varying linear state feedback with feedback gain

$$K(t) := R^{-1}(t)B^T(t)P(t)$$

  although the control problem is formulated to ask for an open loop control.
- The open loop optimal control can be obtained, if so desired, by integrating (1) with the control (12). It is, however, much better to utilize feedback than to use openloop.
- $P(t)$ is solved backwards in time from $t_f \to t_0$ and stored in memory before use.

- The matrix function $P(t)$ is associated with the so-called cost-to-go function. If at time t, $t_0 \leq t \leq t_f$ and the state happens to be $x(t)$, then, the control policy (12) for the remaining time period $[t, t_f]$ is also optimal for the problem (2) $J(u, x(t), t, t_f)$ (i.e. with t_0 substituted by t and x_0 substituted by $x(t)$). In this case, the minimum cost is

$$min_u J(u, x(t), t) = \frac{1}{2} x^T(t) P(t) x(t)$$