4F3 – Predictive Control

Lecture 3

Unconstrained Predictive Control

Dr Eric Kerrigan

Outline Of Course

- Introduction to predictive control
- Digital state space control theory
- Unconstrained predictive control
- Predictive control with constraints
- Set-point tracking and offset-free control
- Stability and feasibility in predictive control - Dr Jan Maciejowski
- Case study by industrial speaker – Dr Paul Austin

Outline Of Part 2:3

- The linear quadratic regulator (LQR) problem
- A finite horizon optimal control problem
- The receding horizon principle revisited (with maths)
- Derivation of prediction matrices
- Derivation of receding horizon control (RHC) law
- Output feedback predictive control
- Equivalence of RHC and LQR
- Alternative formulations
- Introduction to stability issues in predictive control

Recommended Books

- Predictive Control:
  - J.M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, UK (2002)
  - E.F. Camacho and C. Bordons. *Model Predictive Control, 2nd* ed., Springer, UK (2004)
- Digital Control (LQR and Kalman Filtering):
  - G.F. Franklin et al. *Digital Control of Dynamic Systems*, 3rd ed., Addison Wesley (1998)
  - K.J. Åström and B. Wittenmark. *Computer Controlled Systems*, 3rd ed., Prentice Hall (1997)

Standing Assumptions

- For the remainder of the course:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$
$$z(k) = Hx(k)$$

- Unless stated otherwise, we will assume:
  - (A,B) stabilizable and (C,A) detectable
  - $C = I \Rightarrow$ ***state feedback***
  - $H = C \Rightarrow$ all outputs/states are controlled variables (CVs)
  - The goal is to **regulate the states** around the origin
    - Consider time-varying set-points/references and $H \neq C$ in Lecture 5
  - All delays (including computational) are captured in the model
  - No disturbances, noise or model uncertainty (except in Lecture 5)

Linear Quadratic Regulator (LQR) Problem

- Given initial state $x(0)$ at time $k = 0$
- Compute and implement the ***infinite*** input sequence $u(0),u(1),\ldots,$ that minimises the cost:

$$\sum_{k=0}^{\infty} x(k)^T Q x(k) + u(k)^T R u(k)$$

- The designer has to specify the matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$
- The ***state weight*** Q penalises non-zero states
- The ***input weight*** R penalises non-zero control actions
- Q and R are often diagonal and positive definite (but not always)
- Why the term LQR?
  - L = **Linear** model
  - Q = **Quadratic** cost
  - R = **Regulation** around origin

Linear Quadratic Regulator (LQR) Problem

- ***Infinite*** number of decision variables
- Appears to be impossible to solve
- However, a tractable solution exists if:
  - Q is positive semi-definite ($Q \geq 0$)
  - $(Q^{½},A)$ is detectable
  - R is positive definite ($R > 0$)
- Often $x(k)^T Q x(k)$ is replaced by $z(k)^T M z(k)$, where $M \in \mathbb{R}^{q \times q}$:
  - "$(Q^{½},A)$ detectable" then replaced by "$(M^{½}H,A)$ detectable"
- See Mini-tutorial 7 in JMM, any good book on digital and optimal control or 4F2 course notes
- Rather than solving the ***infinite*** horizon LQR problem, we will solve a ***finite*** horizon version
- **NB**: We keep the above assumptions on Q and R

Finite Horizon Optimal Control Problem

- Given current state $x=x(k)$, compute a ***finite*** input sequence $u_0,u_1,...,u_{N-1}$ of length N that minimises:

$$V(x, u_0, \ldots, u_{N-1}) := x_N^T P x_N + \sum_{s=0}^{N-1} x_s^T Q x_s + u_s^T R u_s$$

  where

$$x_0 = x$$
$$x_{s+1} = A x_s + B u_s \qquad s = 0,1,...,N\text{-}1$$

- Note the cost function $V(\cdot)$ is a function only of the current state x and the N inputs $u_0,u_1,...,u_{N-1}$
- **Important**: $V(\cdot)$ is *not* a function of current time k:

  $V(x(j),U) = V(x(k),U)$ if $x(j) = x(k)$ and $j \neq k$

Finite Horizon Optimal Control Problem

- x_s denotes the ***prediction/estimate*** of $x(k+s)$, given current state $x(k)$, if the input sequence is $u(k+i) = u_i$ for all $i = 0,1,\ldots,s-1$
  - JMM uses notation: $\hat{x}(k+s|k) \equiv x_s$, $\hat{u}(k+s|k) \equiv u_s$
  - Our notation results in simpler-looking equations
- N and P affect the **stability** and **performance** – specified by the designer:
  - N is an integer and is called the ***control horizon***
  - Matrix $P \in \mathbb{R}^{n\times n}$ is the ***terminal weight*** and $P \geq 0$

Some Notation

- We define the ***stacked vectors*** $U \in \mathbb{R}^{Nm}$ and $X \in \mathbb{R}^{Nn}$:

$$U := \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix}, \quad X := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- Note that each $u_s \in \mathbb{R}^m$ and $x_s \in \mathbb{R}^n$
- $Y \in \mathbb{R}^{Np}$ and $Z \in \mathbb{R}^{Nq}$ are similarly defined

Some Notation

- The ***cost function*** is defined as:

$$V(x,U) := x_N^T P x_N + \sum_{s=0}^{N-1} x_s^T Q x_s + u_s^T R u_s$$

- The ***value function*** is defined as:

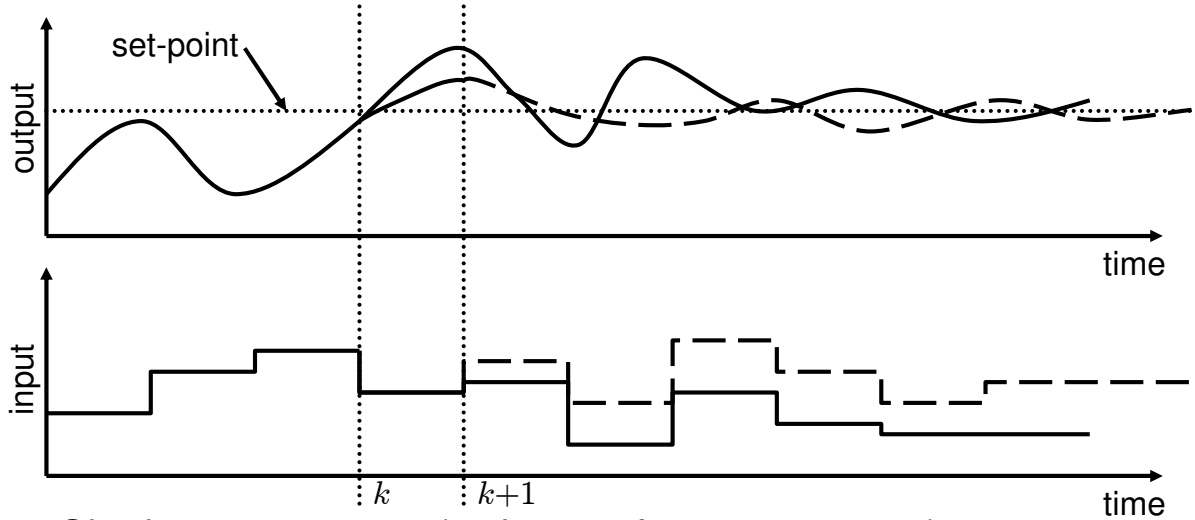
$$V^*(x) := \min_U V(x,U)$$

- The ***optimal input sequence*** is defined as:

$$U^*(x) := \begin{pmatrix} u_0^*(x)^T & \cdots & u_{N-1}^*(x)^T \end{pmatrix}^T$$
$$:= \arg\min_U V(x,U)$$

- Note that above are *not* functions of current time k

The Receding Horizon Principle



1. Obtain measurement/estimate of ***current*** output/state
2. Compute optimal input sequence over a ***finite*** horizon
3. Implement only ***first*** part of input sequence
4. Obtain ***new*** measurement and go to step 2

Unconstrained Predictive Control

- At each time instant k, a predictive controller
  - uses a measurement/estimate of the *current state* x and
  - an internal *model* of the system $x_{s+1} = Ax_s + Bu_s$ to
  - compute a *finite* sequence of control inputs U that
  - minimises the *cost function* $V(x,\cdot)$
- The optimal control sequence is implemented in a ***receding horizon*** fashion:
  - Implement only the *first* input in the optimal sequence:
    $$u(k) = u_0^*(x(k))$$
  - When the optimal control sequence is implemented like this, it is called ***receding horizon control (RHC)***
- The repeated solution of the finite horizon problem "approximates" the infinite horizon problem

Analytical Derivation Of RHC Law

- Compute the **prediction matrices** Φ and Γ in
  $$X = \Phi x + \Gamma U$$
- Rewrite **cost function** $V(\cdot)$ in terms of x and U
- Compute **gradient** of cost function: $\nabla_U V(x,U)$
- Set gradient equal to **zero**: $\nabla_U V(x,U) = 0$
- **Solve** for U to get optimal $U^*(x)$
- The RHC gain matrix $K_{\mathrm{RHC}} \in \mathbb{R}^{m \times n}$ is found from expression for **first input** in optimal sequence:
  $$\begin{aligned} u_0^*(x) &= \begin{pmatrix} I_m & 0_{m\times(N-1)m} \end{pmatrix} U^*(x) \\ &= K_{\mathrm{RHC}}\, x \end{aligned}$$

Derivation Of Prediction Matrices

$$
\begin{aligned}
x_1 &= Ax_0 + Bu_0 \\
x_2 &= A(x_1) + Bu_1 \\
x_3 &= A(x_2) + Bu_2 \\
&\vdots \\
\Rightarrow x_1 &= Ax_0 + Bu_0 \\
x_2 &= A(Ax_0 + Bu_0) + Bu_1 = A^2x_0 + ABu_0 + Bu_1 \\
x_3 &= A^3x_0 + A^2Bu_0 + ABu_1 + Bu_2 \\
&\vdots \\
x_N &= A^Nx_0 + A^{N-1}Bu_0 + \cdots + Bu_{N-1}
\end{aligned}
$$

Derivation Of Prediction Matrices

- Compare with solution on page 14 of Lecture 2
- Collecting terms gives:

$$
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} x_0 + \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix}
$$

- Since $x_0 = x$, the ***prediction matrices*** in $X = \Phi x + \Gamma U$ are:

$$
\Phi := \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix}, \qquad \Gamma := \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{pmatrix}
$$

Rewriting Expression For $V(\cdot)$

$$V(x,U) := x_N^T P x_N + \sum_{s=0}^{N-1} x_s^T Q x_s + u_s^T R u_s$$

$$= x_0^T Q x_0 + \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}^T \begin{pmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & P \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

$$+ \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix}^T \begin{pmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & R \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix}$$

Rewriting Expression For $V(\cdot)$

…getting a little bit dull…

- Recalling $x_0 = x$ and definitions of X and U gives:

$$V(x,U) = x^T Q x + X^T \Omega X + U^T \Psi U$$

where the square matrices Ω and Ψ are:

$$\Omega := \begin{pmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & P \end{pmatrix}, \qquad \Psi := \begin{pmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & R \end{pmatrix}$$

- Note that:
  - $P \geq 0$ and $Q \geq 0 \Rightarrow \Omega \geq 0$
  - $R > 0 \Rightarrow \Psi > 0$

Rewriting Expression For $V(\cdot)$ …now I've really gone to sleep…

Finally, recalling that $X = \Phi x + \Gamma U$, we get

$$\begin{aligned}
V(x,U) &= U^T\Psi U + X^T\Omega X + x^TQx \\
&= U^T\Psi U + (\Phi x + \Gamma U)^T\Omega(\Phi x + \Gamma U) + x^TQx \\
&= U^T\Psi U + (x^T\Phi^T + U^T\Gamma^T)(\Omega\Phi x + \Omega\Gamma U) + x^TQx \\
&= U^T\Psi U + x^T\Phi^T(\Omega\Phi x + \Omega\Gamma U) \\
&\quad + U^T\Gamma^T(\Omega\Phi x + \Omega\Gamma U) + x^TQx \\
&= U^T\Psi U + x^T\Phi^T\Omega\Phi x + x^T\Phi^T\Omega\Gamma U \\
&\quad + U^T\Gamma^T\Omega\Phi x + U^T\Gamma^T\Omega\Gamma U + x^TQx \\
&= U^T\Psi U + x^T\Phi^T\Omega\Phi x + U^T\Gamma^T\Omega\Phi x \\
&\quad + U^T\Gamma^T\Omega\Phi x + U^T\Gamma^T\Omega\Gamma U + x^TQx \\
&= U^T(\Psi + \Gamma^T\Omega\Gamma)U + 2U^T\Gamma^T\Omega\Phi x + x^T(Q + \Phi^T\Omega\Phi)x
\end{aligned}$$

Computing Gradient Of $V(\cdot)$

- After some easy (but tedious) linear algebra, we get that
  $$V(x,U) = (1/2)U^TGU + U^TFx + x^T(Q + \Phi^T\Omega\Phi)x$$
  where the (square) matrix G is
  $$G := 2(\Psi + \Gamma^T\Omega\Gamma)$$
  and the matrix F is
  $$F := 2\Gamma^T\Omega\Phi$$
- Recall:
  - Quadratic function: $q(U) = (1/2)\, U^TGU + U^Tf + c$, where G is symmetric
  - Gradient is $\nabla_U q(U) = GU + f$ (written as a column vector)
- The ***gradient*** of the cost function is therefore:
  $$\nabla_U V(x,U) = GU + Fx$$
  - Note the gradient ***is*** a function of $f := Fx$,
  - but the gradient ***is not*** a function of $c := x^T(Q + \Phi^T\Omega\Phi)x$

Expression For Optimal U

- Set the **gradient** of the cost function to 0:
$$\nabla_U V(x,U) = GU + Fx = 0$$
- Solve for U in the set of **linear equalities**:
$$GU = -Fx$$
- Recall that $\Psi > 0$ and $\Omega \geq 0$:
$\Rightarrow G := \Psi + \Gamma^T \Omega \Gamma > 0$
$\Rightarrow G$ is ***invertible***
- The above two facts imply that for any x, a solution (U) to $GU = -Fx$
  - ***exists***,
  - is ***unique*** and
  - is a ***global minimiser*** of $V(x,\cdot)$
- Expression for the optimal input sequence is therefore:
$$U^*(x) = -G^{-1}Fx$$

Expression For RHC Law

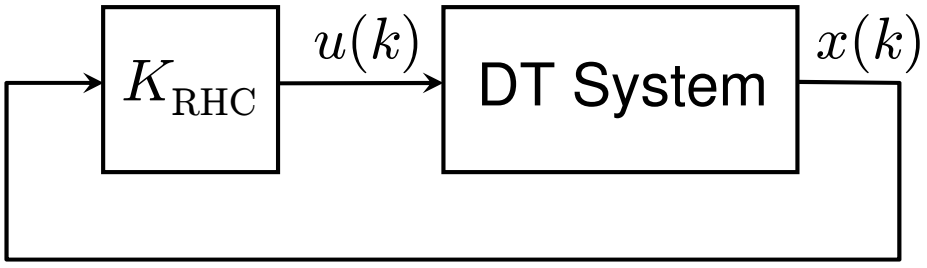
- The RHC gain matrix $K_{\mathrm{RHC}} \in \mathbb{R}^{m \times n}$ is found from first input in optimal sequence:
$$\begin{aligned} u_0^*(x) &= \begin{pmatrix} I_m & 0_{m \times (N-1)m} \end{pmatrix} U^*(x) \\ &= K_{\mathrm{RHC}}\, x \end{aligned}$$
- Expression for optimal input sequence is:
$$U^*(x) = -G^{-1}Fx$$
- Hence,
$$K_{\mathrm{RHC}} := -\begin{pmatrix} I_m & 0_{m \times (N-1)\,m} \end{pmatrix} G^{-1} F$$

Again: Analytical Derivation Of RHC Law

- Compute the **prediction matrices** Φ and Γ in
$$X = \Phi x + \Gamma U$$
- Rewrite **cost function** $V(\cdot)$ in terms of x and U
- Compute **gradient** of cost function: $\nabla_U V(x,U)$
- Set gradient equal to **zero**: $\nabla_U V(x,U) = 0$
- **Solve** for U to get optimal $U^*(x)$
- The RHC gain matrix $K_{\mathrm{RHC}} \in \mathbb{R}^{m \times n}$ is found from expression for **first input** in optimal sequence:
$$\begin{aligned} u_0^*(x) &= \begin{pmatrix} I_m & 0_{m\times(N-1)m} \end{pmatrix} U^*(x) \\ &= K_{\mathrm{RHC}}\, x \end{aligned}$$

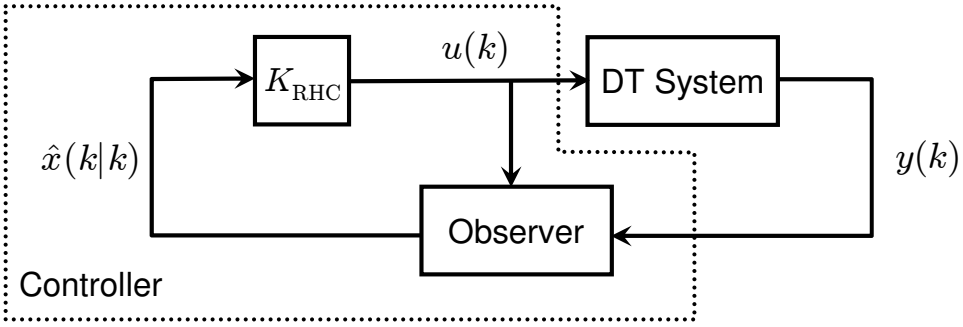
Implementation Of RHC Law

- Note that $K_{\mathrm{RHC}} \in \mathbb{R}^{m \times n}$ is a ***time-invariant***, linear, state feedback gain
- RHC law is given by $u(k) = K_{\mathrm{RHC}} x(k)$



Output Feedback Predictive Control

- $C \neq I \Rightarrow$ output feedback
- Use an **observer** or **Kalman filter** to provide an estimate of the state $\hat{x}(k|k)$
- Computation and implementation of RHC law is the same, except with $x(k)$ replaced with $\hat{x}(k|k)$



Equivalence Between RHC And LQR

- The solution to the ***infinite-horizon*** LQR problem is:

$$u(k) = K_{\mathrm{LQR}} x(k)$$

where

$$K_{\mathrm{LQR}} = -(B^T P B + R)^{-1} B^T P A$$

and P is the solution to the ***Algebraic Riccati Equation (ARE)***:

$$P = A^T P A - A^T P B (B^T P B + R)^{-1} B^T P A + Q$$

- See 4F2 notes or JMM Mini-tutorial 7 for derivation

- If the ***terminal weight*** P in the ***finite-horizon*** cost function $V(\cdot)$ is the solution to above ARE, then

$$K_{\mathrm{RHC}} = K_{\mathrm{LQR}}$$

- This is an important precursor in solving the problem of ***infinite-horizon*** LQR ***with constraints***

Alternative Formulations

- Many other variations exist on the predictive control formulation presented in this course
- One common formulation is to optimise over predicted ***changes*** $\Delta u_s := u_s - u_{s-1}$, rather than u_s
  - The cost function $V(\cdot)$ is then a function of $x(k)$, the decision variables $\Delta u_0, \Delta u_1, \ldots, \Delta u_{N-1}$ ***and*** $u(k-1)$
- Another common formulation is to have a larger horizon for prediction than for control
  - The inputs are then often constrained after the end of the control horizon, e.g. $u_s = Kx_s$ or $\Delta u_s = 0$ for all $s \geq N$

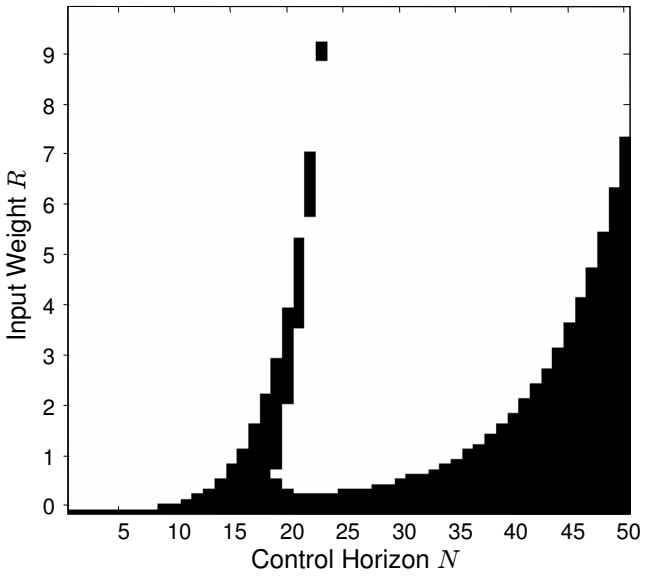
Stability In Predictive Control

- **Warning**: The RHC law $u(k) = K_{\mathrm{RHC}}\, x(k)$ is not necessarily stabilizing
- How to choose a Q, R, P and N in order to guarantee stability is not always obvious
- Consider, for example, the following open-loop unstable system:

$$x(k+1) = \begin{pmatrix} 1.216 & -0.055 \\ 0.221 & 0.9947 \end{pmatrix} x(k) + \begin{pmatrix} 0.02763 \\ 0.002763 \end{pmatrix} u(k)$$

Stability In Predictive Control

- Fix $Q = P = I$
- Compute $\rho(A+BK_{\mathrm{RHC}})$ for various N and R:
  - Vary $N = 1,2,\ldots,50$
  - Vary $R = 0,0.2,\ldots,10$
- Plot:
  - Black – stable, i.e. $\rho(A+BK_{\mathrm{RHC}}) < 1$
  - White – unstable, i.e. $\rho(A+BK_{\mathrm{RHC}}) \geq 1$



- Graph shows that not all values of N and R guarantee a stable closed-loop – this is a problem we will address in Lecture 6

Summary

- Posed and solved a **finite horizon** version of the LQR problem
- Receding horizon principle:
  - Compute optimal input sequence at **each time** instant
  - Only **implement first input** of the optimal sequence
  - Repeated solution of a finite horizon problem "approximates" the infinite horizon problem
- Analytical derivation of RHC law without constraints:
  - Obtain an expression for **cost function** $V(\cdot)$ in terms of x and U only
  - Compute **gradient** of cost function $\nabla_U V(x,U)$ and set to **zero**
  - Optimal input sequence is solution to resulting set of linear equalities
  - The RHC law is a linear, time-invariant, **state feedback gain matrix**
- The RHC law can be made to be the same as the LQR law by appropriate choice of terminal weight P
- Arbitrary choice of weights P, Q and R and control horizon N could lead to a RHC closed-loop system that is **unstable**