Linear regulation and state estimation (LQR and LQE)

James B. Rawlings

Department of Chemical and Biological Engineering

THE UNIVERSITY of WISCONSIN MADISON

Freiburg, Germany
March 30-31, 2015

Outline

Linear differential equations

The linear continuous time model is a set of n first-order linear differential equations

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{1}$$

in which

x	state	n-vector
u	manipulated input	m-vector
y	measured output	p-vector

Discrete time linear difference equations

- For discrete time systems, we have a sample time, denoted Δ, and we are interested in the state, input and output of the system only at the sample times, $t = k\Delta$.
- The linear difference equation that represents the behavior at the sample times is given by

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned} \tag{2}$$

- The following formulas let us convert from the continuous time differential equation model to the discrete time model difference equation model

$$A_d = e^{A\Delta} \qquad B_d = A^{-1}(e^{A\Delta} - I_n)B \qquad C_d = C \qquad D_d = D \tag{3}$$

Formulas are exact!

- Note that if $u(t)$ is a constant between samples (known as a zero-order hold), then these formulas are exact and there is no approximation error such as we would have if we used an Euler method to approximate the time derivative in (1).
- Note also that if A is singular (one or more of its eigenvalues are zero), A^{-1} does not exist and the formula for B_d is not defined.
- You can obtain B_d from the following relationship that does not require A^{-1}

$$\exp\left(\Delta \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}$$

A point on dot and + notation

Just like in the continuous time model where we use the dot notation to represent the time derivative

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu & \qquad \dot{x} &= Ax + Bu \\ y &= Cx + Du & \qquad y &= Cx + Du \end{aligned}$$

In discrete time we use the + notation to represent the state at the next sample time

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) & \qquad x^{+} &= Ax + Bu \\ y(k) &= Cx(k) + Du(k) & \qquad y &= Cx + Du \end{aligned}$$

Linear quadratic regulation

- We start by designing a controller to take the state of a deterministic, linear system to the origin.
- If the setpoint is not the origin, or we wish to track a time-varying setpoint trajectory, we will subsequently make modifications of the zero setpoint problem to account for that.
- The system model is

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx \end{aligned} \tag{4}$$

in which x is an n vector, u is an m vector, and y is a p vector.

State feedback

- In this first problem, we assume that the state is measured, or $C = I$.
- We will handle the output measurement problem with state estimation in the next section.
- Using the model we can predict how the state evolves given any set of inputs we are considering.

Sequence of inputs (decision variables)

- Consider N time steps into the future and collect the input sequence into $\mathbf{u}$,

$$\mathbf{u} = \{u(0), u(1), \ldots, u(N-1)\}$$

- Constraints on the $\mathbf{u}$ sequence (i.e., valve saturations, etc.) are the main feature that distinguishes MPC from the standard linear quadratic (LQ) control.

Constraints

- The manipulated inputs (valve positions, voltages, torques, etc.) to most physical systems are bounded. We include these constraints by linear inequalities

$$Eu(k) \leq e \qquad k = 0, 1, \ldots$$

in which

$$E = \begin{bmatrix} I \\ -I \end{bmatrix} \qquad e = \begin{bmatrix} \overline{u} \\ -\underline{u} \end{bmatrix}$$

are chosen to describe simple bounds such as

$$\underline{u} \leq u(k) \leq \overline{u} \qquad k = 0, 1, \ldots$$

- We sometimes wish to impose constraints on states or outputs for reasons of safety, operability, product quality, etc. These can be stated as

$$Fx(k) \leq f \qquad k = 0, 1, \ldots$$

Rate of change constraints

- Practitioners find it convenient in some applications to limit the rate of change of the input, $u(k) - u(k-1)$. To maintain the state space form of the model, we may augment the state as

$$\tilde{x}(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}$$

and the augmented system model becomes

$$\begin{aligned} \tilde{x}^+ &= \tilde{A}\tilde{x} + \tilde{B}u \\ y &= \tilde{C}\tilde{x} \end{aligned}$$

in which

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix} \qquad \tilde{C} = \begin{bmatrix} C & 0 \end{bmatrix}$$

Rate of change constraints

- A rate of change constraint such as

$$\underline{\Delta} \leq u(k) - u(k-1) \leq \overline{\Delta} \qquad k = 0, 1, \ldots$$

is then stated as

$$F\tilde{x}(k) + Eu(k) \leq e \qquad F = \begin{bmatrix} 0 & -I \\ 0 & I \end{bmatrix} \qquad E = \begin{bmatrix} I \\ -I \end{bmatrix} \qquad e = \begin{bmatrix} \overline{\Delta} \\ -\underline{\Delta} \end{bmatrix}$$

- To simplify analysis, it pays to maintain linear constraints when using linear dynamic models. So if we want to consider fairly general constraints for a linear system, we choose the form

$$Fx(k) + Eu(k) \leq e \qquad k = 0, 1, \ldots$$

which subsumes all the forms listed previously.

Controller objective function

- We first define an objective function $V(\cdot)$ to measure the deviation of the trajectory of $x(k), u(k)$ from zero by summing the weighted squares

$$V(x(0), \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{N-1} \left[x(k)' Q x(k) + u(k)' R u(k) \right] + \frac{1}{2} x(N)' P_f x(N)$$

subject to

$$x^+ = Ax + Bu$$

- The objective function depends on the input sequence and state sequence.
- The initial state is available from the measurement. The remainder of the state trajectory, $x(k), k = 1, \ldots, N$, is determined by the model and the input sequence $\mathbf{u}$.
- So we show the objective function's explicit dependence on the input sequence and *initial* state.

Regulator tuning parameters

- The tuning parameters in the controller are the matrices Q and R. Note that we are using the deviation of u from zero as the input penalty. We'll use rate of change penalty S later.
- We allow the final state penalty to have a different weighting matrix, P_f, for generality.
- Large values of Q in comparison to R reflect the designer's intent to drive the state to the origin quickly at the expense of large control action.
- Penalizing the control action through large values of R relative to Q is the way to reduce the control action and slow down the rate at which the state approaches the origin.
- Choosing appropriate values of Q and R (i.e., tuning) is not always obvious, and this difficulty is one of the challenges faced by industrial practitioners of LQ control. Notice that MPC inherits this tuning challenge.

Regulation problem—definition

- We then formulate the following optimal MPC control problem

$$\min_{\mathbf{u}} V(x(0), \mathbf{u}) \tag{5}$$

subject to

$$x^+ = Ax + Bu \qquad Fx(k) + Eu(k) \leq e, \quad k = 0, 1, \ldots, N-1$$

- The Q, P_f and R matrices often are chosen to be diagonal, but we do not assume that here. We assume, however, that Q, P_f, and R are *real and symmetric*; Q and P_f are *positive semidefinite*; and R is *positive definite*.
- These assumptions guarantee that the solution to the optimal control problem exists and is unique.

Let's first examine a scalar system ($n = 1$) with horizon $N = 1$

- We have only one move to optimize, u_0.

$$x_1 = ax_0 + bu_0$$

- The objective is

$$V = (1/2)\big(qx_0^2 + ru_0^2 + p_f x_1^2\big)$$

- Expand the x_1 term to see its dependence on the control u_0

$$\begin{aligned} V &= (1/2)\big(qx_0^2 + ru_0^2 + p_f(ax_0 + bu_0)^2\big) \\ &= (1/2)\Big(qx_0^2 + ru_0^2 + p_f(a^2x_0^2 + 2abx_0u_0 + b^2u_0^2)\Big) \\ V &= (1/2)\Big((q + a^2p_f)x_0^2 + 2(bap_f x_0)u_0 + (b^2p_f + r)u_0^2\Big) \end{aligned}$$

Take derivative dV/du_0, set to zero

- Take the derivative, set to zero

$$\begin{aligned}\frac{d}{du_0}V &= bp_f a x_0 + (b^2 p_f + r)u_0 \\ 0 &= bp_f a x_0 + (b^2 p_f + r)u_0\end{aligned}$$

- Solve for optimal control

$$\begin{aligned}u_0^0 &= -\frac{bp_f a}{b^2 p_f + r} x_0 \\ u_0^0 &= k x_0 \qquad k = -\frac{bp_f a}{b^2 p_f + r}\end{aligned}$$

Back to vectors and matrices

- Quadratic functions in x_0 and u_0

$$V = (1/2)\left(x_0'(Q + A'P_f A)x_0 + 2u_0' B' P_f A x_0 + u_0'(B'P_f B + R)u_0 \right)$$

- Take derivative, set to zero, solve

$$\begin{aligned} \frac{d}{du_0} V &= B'P_f A x_0 + (B'P_f B + R)u_0 \\ u_0^0 &= -(B'P_f B + R)^{-1} B'P_f A x_0 \end{aligned}$$

- So we have the optimal linear control law

$$u_0^0 = K x_0 \qquad K = -(B'P_f B + R)^{-1} B' P_f A$$

Dynamic programming solution

- To handle $N > 1$, we can go to the end of the horizon and work our way backwards to the beginning of the horizon.
- This trick is known as (backwards) dynamic programming. Developed by Bellman (1957).
- Each problem that we solve going backwards is an $N = 1$ problem, which we already know how to solve!
- When we arrive at $k = 0$, we have the optimal control as a function of the initial state. This is our control law.
- We next examine the outcome of applying dynamic programming to the LQ problem.

The regulator Riccati equation

- The recursion from $\Pi(k)$ to $\Pi(k-1)$ is known as a backward Riccati iteration. To summarize, the backward Riccati iteration is defined as follows

$$\Pi(k-1) = Q + A'\Pi(k)A - A'\Pi(k)B\left(B'\Pi(k)B + R\right)^{-1}B'\Pi(k)A \qquad k = N, N-1, \ldots, 1 \quad (6)$$

with terminal condition

$$\Pi(N) = P_f \quad (7)$$

- The terminal condition replaces the typical initial condition because the iteration is running backward.

The optimal control policy

- The optimal control policy at each stage is

$$u_k^0(x) = K(k)x \qquad k = N-1, N-2, \dots, 0 \tag{8}$$

- The optimal gain at time k is computed from the Riccati matrix at time $k+1$

$$K(k) = -\left(B'\Pi(k+1)B + R\right)^{-1} B'\Pi(k+1)A \qquad k = N-1, N-2, \dots, 0 \tag{9}$$

- The optimal cost to go from time k to time N is

$$V_k^0(x) = (1/2)x'\Pi(k)x \qquad k = N, N-1, \dots, 0 \tag{10}$$

Whew. OK, the controller is optimal, but is it useful?

- We now have set up and solved the (unconstrained) optimal control problem.
- Next we are going to study some of the closed-loop properties of the controlled system under this controller. One of these properties is closed-loop stability.
- Before diving into the controller stability discussion, let's take a breather and see what kinds of things that we should guard against when using the optimal controller.
- We start with a simple scalar system with inverse response or, equivalently, a right-half-plane zero in the system transfer function.

$$y(s) = g(s)u(s) \qquad g(s) = k\frac{s-a}{s+b}, \quad a, b > 0$$

Step response of system with right-half-plane zero

- Convert the transfer function to state space (notice a nonzero D)

$$\frac{d}{dt}x = Ax + Bu \qquad A = -b \qquad B = -(a+b)$$
$$y = Cx + Du \qquad C = k \qquad D = k$$

- Solve the ODEs with $u(t) = 1$ and $x(0) = 0$

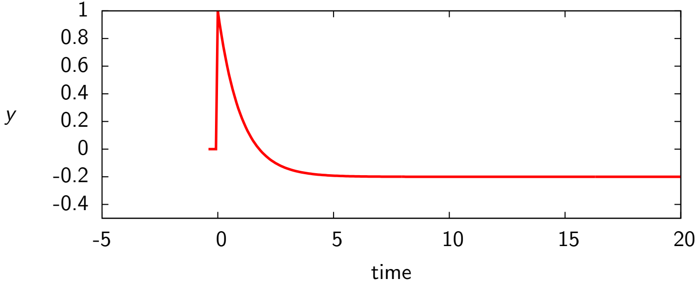


Figure 1: Step response of a system with a right-half-plane zero.

What input delivers a unit step change in the output?

- Say I make a unit step change in the setpoint of y. What $u(t)$ (if any) can deliver a (perfect) unit step in $y(t)$?
- In the Laplace domain, $\overline{y}(s) = 1/s$. Solve for $\overline{u}(s)$. Since $\overline{y} = g(s)\overline{u}$ we have that

$$\overline{u}(s) = \frac{\overline{y}(s)}{g(s)} = \frac{s+b}{ks(s-a)}$$

- We can invert this signal back to the time domain[a] to obtain

$$u(t) = \frac{1}{ka}\Big[-b + (a+b)e^{at}\Big]$$

- Note that the second term is a growing exponential for $a > 0$. Hmmm... How well will that work?

[a] Use partial fractions, for example.

Simulating the system with chosen input

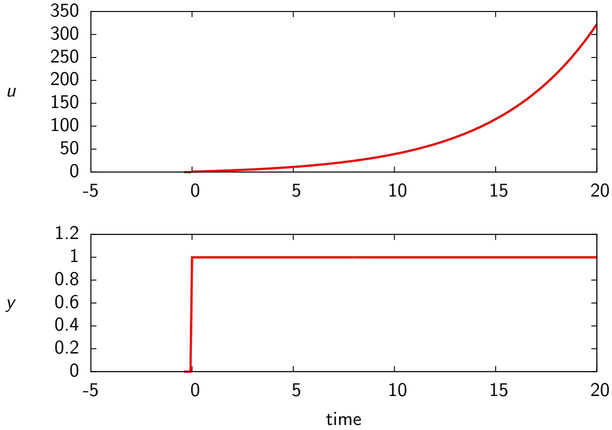


Figure 2: Output response with an exponential input for a system with RHP zero.

- Sure enough, that input moves the output right to its setpoint!

Will this work in practice?

- So we can indeed achieve perfect tracking in $y(t)$ with this growing input $u(t)$.
- Because the system $g(s) = k(s-a)/(s+b)$ has a zero at $s = a$, and $\overline{u}(s)$ has a $1/(s-a)$ term, they cancel and we see nice behavior rather than exponential growth in $y(t)$.
- This cancellation is the so-called input blocking property of the transfer function zero.
- But what happens if $u(t)$ hits a constraint, $u(t) \leq u_{\max}$.
- Let's simulate the system again, but with an input constraint at $u_{\max} = 20$. We achieve the following result.

Output behavior when the exponential input saturates

- The input saturation destroys the nice output behavior

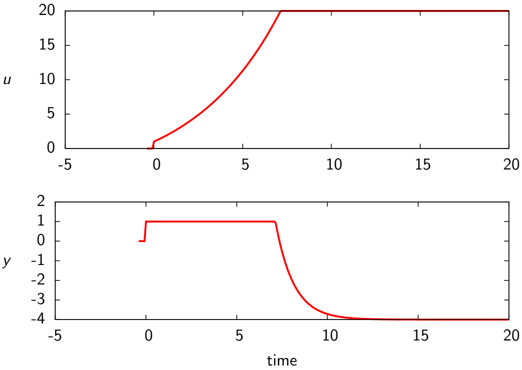


Figure 3: Output response with input saturation for a system with RHP zero.

So how can we reliably use optimal control?

Control theory to the rescue

- The optimal controller needs to know to avoid exploiting this exponential input. But how do we tell it?
- We cannot let it start down the exponential path and find out only later that this was a bad idea.
- We have to eliminate this option by the structure of the optimal control problem.
- More importantly, even if we fix this issue, how do we know that we have eliminated every other path to instability for every other kind of (large, multivariable) system?
- Addressing this kind of issue is where control theory is a lot more useful than brute force simulation.

The infinite horizon LQ problem

- Let us motivate the infinite horizon problem by showing a weakness of the finite horizon problem. Kalman (1960b, p.113) pointed out in his classic 1960 paper that optimality does not ensure stability.

  > *In the engineering literature it is often assumed (tacitly and incorrectly) that a system with an optimal control law is necessarily stable.*

- Assume that we use as our control law the first feedback gain of the N stage finite horizon problem, $K_N(0)$,

$$u(k) = K_N(0)x(k)$$

- Then the stability of the closed-loop system is determined by the eigenvalues of $A + BK_N(0)$. We now construct an example that shows choosing $Q > 0$, $R > 0$, and $N \geq 1$ does not ensure stability. In fact, we can find reasonable values of these parameters such that the controller destabilizes a stable system.

An optimal controller that is closed-loop unstable

Let

$$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = [-2/3 \;\; 1]$$

- This discrete-time system has a transfer function *zero* at $z = 3/2$, i.e., an unstable zero (or a right-half-plane zero in continuous time).
- We now construct an LQ controller that inverts this zero and hence produces an unstable system.
- We would like to choose $Q = C'C$ so that y itself is penalized, but that Q is only semidefinite.
- We add a small positive definite piece to $C'C$ so that Q is positive definite, and choose a *small* positive R penalty (to encourage the controller to misbehave), and $N = 5$,

$$Q = C'C + 0.001I = \begin{bmatrix} 4/9 + .001 & -2/3 \\ -2/3 & 1.001 \end{bmatrix} \qquad R = 0.001$$

Compute the control law

- We now iterate the Riccati equation four times starting from $\Pi = P_f = Q$ and compute $K_N(0)$ for $N = 5$.
- Then we compute the eigenvalues of $A + BK_N(0)$ and achieve[1]

$$\mathrm{eig}(A + BK_5(0)) = \{1.307, 0.001\}$$

- Using this controller the closed-loop system evolution is $x(k) = (A + BK_5(0))^k x_0$.
- Since an eigenvalue of $A + BK_5(0)$ is greater than unity, $x(k) \to \infty$ as $k \to \infty$.
- In other words the closed-loop system is unstable.

[1] Please check this answer with Octave or MATLAB.

Closed-loop eigenvalues for different N; Riccati iteration

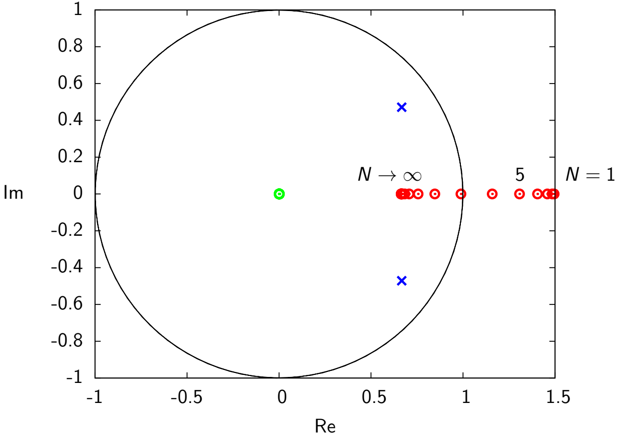


Figure 4: Closed-loop eigenvalues of $A + BK_N(0)$ for different horizon length N (o); open-loop eigenvalues of A (x); real versus imaginary parts.

Stability and horizon length—another view

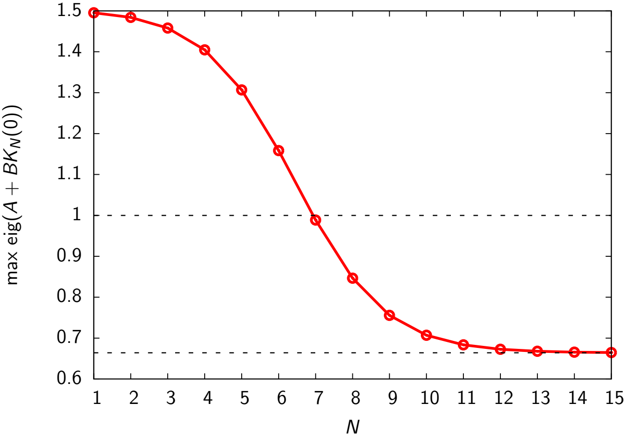


Figure 5: Stability dependence on horizon. The horizontal line at 1 is the stability boundary, while the horizontal line at 0.664 is the maximum eigenvalue of the infinite-horizon regulator.

Why is the closed-loop unstable?

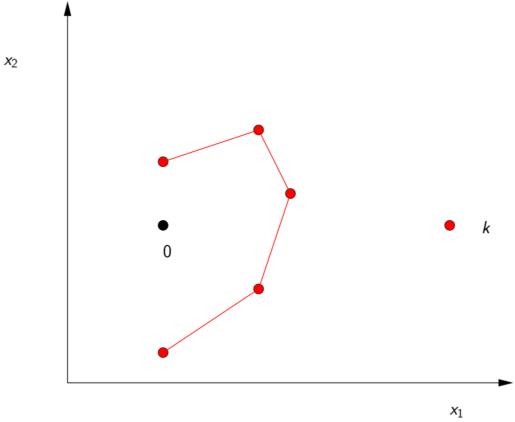
- A finite horizon objective function may not give a stable controller!

Why is the closed-loop unstable?

- A finite horizon objective function may not give a stable controller!
- How is this possible?

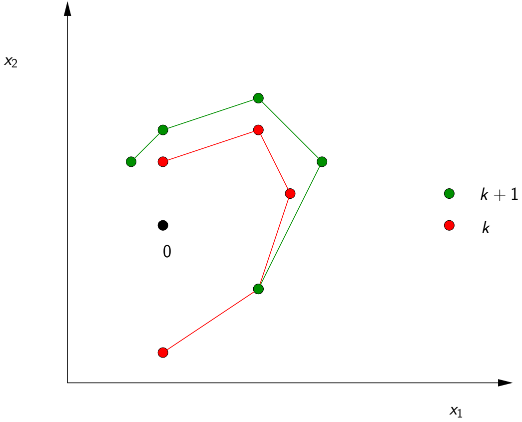
Why is the closed-loop unstable?

- A finite horizon objective function may not give a stable controller!
- How is this possible?



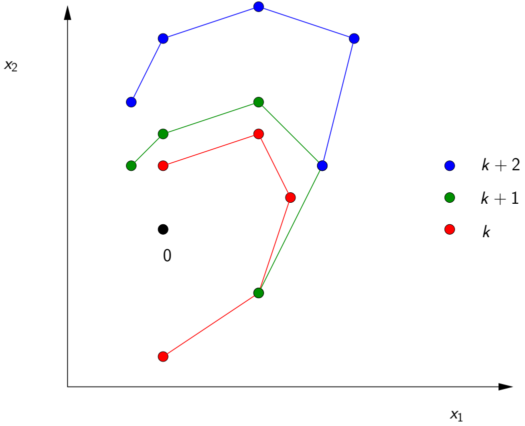
Why is the closed-loop unstable?

- A finite horizon objective function may not give a stable controller!
- How is this possible?



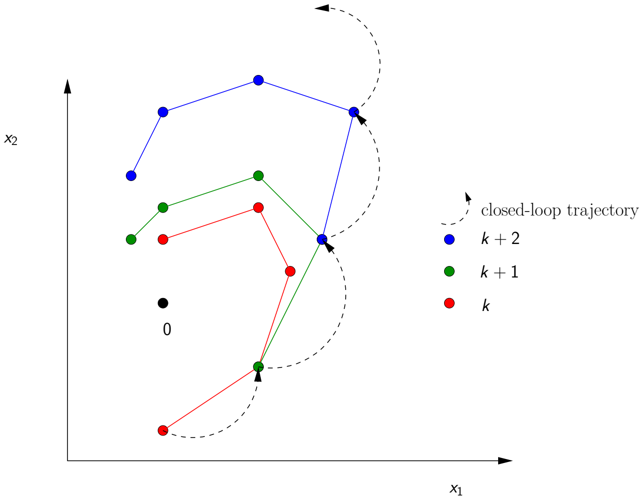
Why is the closed-loop unstable?

- A finite horizon objective function may not give a stable controller!
- How is this possible?



Why is the closed-loop unstable?

- A finite horizon objective function may not give a stable controller!
- How is this possible?



What happens if we make the horizon N larger?

- If we continue to iterate the Riccati equation, which corresponds to increasing the horizon in the controller, we obtain for $N = 7$

$$\mathrm{eig}(A + BK_7(0)) = \{0.989, 0.001\}$$

  and the controller is stabilizing.

- If we continue iterating the Riccati equation, we converge to the following steady-state closed-loop eigenvalues

$$\mathrm{eig}(A + BK_\infty(0)) = \{0.664, 0.001\}$$

- This controller corresponds to an infinite horizon control law.
- Notice that it is stabilizing and has a reasonable stability margin.
- Nominal stability is a guaranteed property of infinite horizon controllers as we prove in the next section.

The forecast versus the closed-loop behavior for $N = 5$

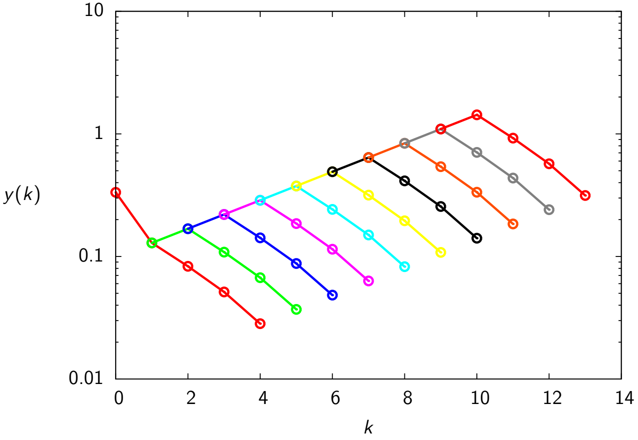


Figure 6: The forecast versus the closed-loop behavior for $N = 5$.

The forecast versus the closed-loop behavior for $N = 7$

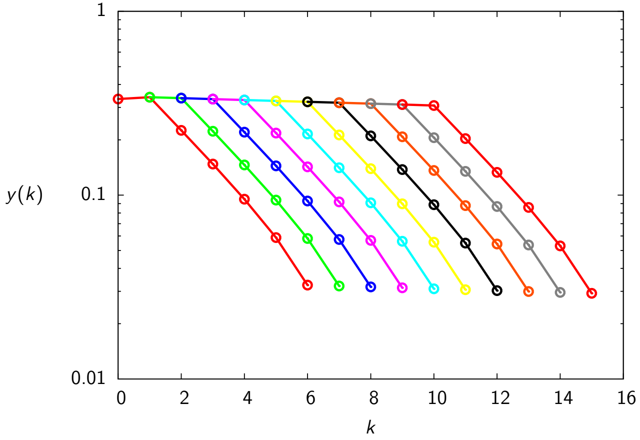


Figure 7: The forecast versus the closed-loop behavior for $N = 7$.

The forecast versus the closed-loop behavior for $N = 20$

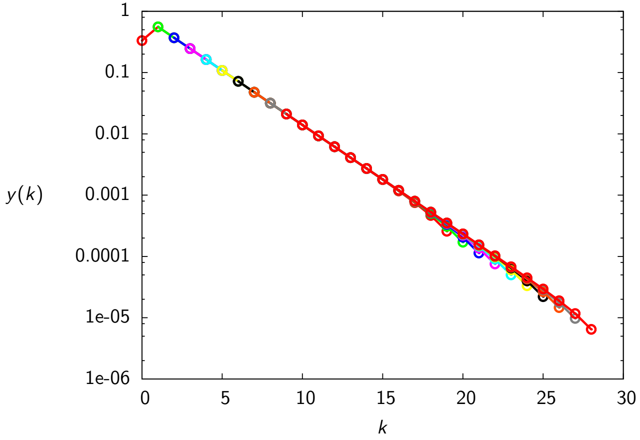


Figure 8: The forecast versus the closed-loop behavior for $N = 20$.

The infinite horizon regulator

- With this motivation, we are led to consider directly the infinite horizon case

$$V(x(0), \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{\infty} x(k)' Q x(k) + u(k)' R u(k) \tag{11}$$

  in which $x(k)$ is the solution at time k of $x^+ = Ax + Bu$ if the initial state is $x(0)$ and the input sequence is $\mathbf{u}$.
- If we are interested in a continuous process (i.e., no final time), then the natural cost function is an infinite horizon cost.
- If we were truly interested in a batch process (i.e., the process does stop at $k = N$), then stability is not a relevant property, and we naturally would use the finite horizon LQ controller and the *time-varying* controller, $u(k) = K(k)x(k), k = 0, 1, \ldots, N$.

When can we compute an infinite horizon cost?

- In considering the infinite horizon problem, we first restrict attention to systems for which there exist input sequences that give bounded cost.
- Consider the case $A = I$ and $B = 0$, for example. Regardless of the choice of input sequence, (11) is unbounded for $x(0) \neq 0$.
- It seems clear that we are not going to stabilize an unstable system ($A = I$) without any input ($B = 0$).
- This is an example of an *uncontrollable* system.
- In order to state the sharpest results on stabilization, we require the concepts of controllability, stabilizability, observability, and detectability. We shall define these concepts subsequently.

Controllability

- A system is *controllable* if, for any pair of states x, z in the state space, z can be reached in finite time from x (or x controlled to z) (Sontag, 1998, p.83).
- A *linear discrete time* system $x^+ = Ax + Bu$ is therefore controllable if there exists a finite time N and a sequence of inputs

$$\{u(0), u(1), \dots u(N-1)\}$$

that can transfer the system from any x to any z in which

$$z = A^N x + \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u(N-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix}$$

Checking with $N = n$ moves is sufficient

- For an unconstrained linear system, if we cannot reach z in n moves, we cannot reach z in any number of moves.
- The question of *controllability* of a linear time-invariant system is therefore a question of *existence* of solutions to linear equations for an arbitrary right-hand side

$$\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix} = z - A^n x$$

- The matrix appearing in this equation is known as the *controllability matrix* $\mathcal{C}$

$$\mathcal{C} = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \tag{12}$$

Controllability test

- From the fundamental theorem of linear algebra, we know a solution exists for all right-hand sides if and only if the *rows* of the $n \times nm$ controllability matrix are linearly independent.[2]
- Therefore, the system (A, B) is controllable if and only if

$$\mathrm{rank}(\mathcal{C}) = n$$

[2] See Section A.4 in Appendix A of (Rawlings and Mayne, 2009) or (Strang, 1980, pp.87–88) for a review of this result.

Convergence of the linear quadratic regulator

- We now show that the infinite horizon regulator asymptotically stabilizes the origin for the closed-loop system. Define the infinite horizon objective function

$$V(x, \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{\infty} x(k)' Q x(k) + u(k)' R u(k)$$

subject to

$$\begin{aligned} x^+ &= Ax + Bu \\ x(0) &= x \end{aligned}$$

with $Q, R > 0$.

Convergence of the linear quadratic regulator

- If (A, B) is controllable, the solution to the optimization problem

$$\min_{\mathbf{u}} V(x, \mathbf{u})$$

  exists and is unique for all x.
- We denote the optimal solution by $\mathbf{u}^0(x)$, and the first input in the optimal sequence by $u^0(x)$.
- The feedback control law $\kappa_\infty(\cdot)$ for this infinite horizon case is then defined as $u = \kappa_\infty(x)$ in which $\kappa_\infty(x) = u^0(x) = \mathbf{u}^0(0; x)$.

LQR stability

As stated in the following lemma, this infinite horizon linear quadratic regulator (LQR) is stabilizing.

Lemma 1 (LQR convergence)

For (A, B) controllable, the infinite horizon LQR with $Q, R > 0$ gives a convergent closed-loop system

$$x^+ = Ax + B\kappa_\infty(x)$$

Controllability gives finite cost

- The cost of the infinite horizon objective is bounded above for all $x(0)$ because (A, B) is controllable.
- Controllability implies that there exists a sequence of n inputs $\{u(0), u(1), \ldots, u(n-1)\}$ that transfers the state from any $x(0)$ to $x(n) = 0$.
- A zero control sequence after $k = n$ for $\{u(n+1), u(n+2), \ldots\}$ generates zero cost for all terms in V after $k = n$, and the objective function for this infinite control sequence is therefore finite.
- The cost function is strictly convex in $\mathbf{u}$ because $R > 0$ so the solution to the optimization is unique.

Proof of closed-loop convergence

- If we consider the sequence of costs to go along the closed-loop trajectory, we have for $V_k = V^0(x(k))$

$$V_{k+1} = V_k - (1/2)\left(x(k)'Qx(k) + u(k)'Ru(k)\right)$$

in which $V_k = V^0(x(k))$ is the cost at time k for state value $x(k)$ and $u(k) = u^0(x(k))$ is the optimal control for state $x(k)$.

- The cost along the closed-loop trajectory is nonincreasing and bounded below (by zero). Therefore, the sequence $\{V_k\}$ converges and

$$x(k)'Qx(k) \to 0 \qquad u(k)'Ru(k) \to 0 \qquad \text{as } k \to \infty$$

- Since $Q, R > 0$, we have

$$x(k) \to 0 \qquad u(k) \to 0 \qquad \text{as } k \to \infty$$

and closed-loop convergence is established.

Connection to Riccati equation

- In fact we know more. From the previous sections, we know the optimal solution is found by iterating the Riccati equation, and the optimal infinite horizon control law and optimal cost are given by

$$u^0(x) = Kx \qquad V^0(x) = (1/2)x'\Pi x$$

in which

$$\begin{aligned} K &= -(B'\Pi B + R)^{-1}B'\Pi A \\ \Pi &= Q + A'\Pi A - A'\Pi B(B'\Pi B + R)^{-1}B'\Pi A \end{aligned} \tag{13}$$

- Proving Lemma 1 has shown also that for (A, B) controllable and $Q, R > 0$, a positive definite solution to the discrete algebraic Riccati equation (DARE), (13), exists and the eigenvalues of $(A + BK)$ are asymptotically stable for the K corresponding to this solution (Bertsekas, 1987, pp.58–64).

Extensions to constrained systems

- This basic approach to establishing regulator stability will be generalized to handle constrained (and nonlinear) systems
- The optimal cost is a Lyapunov function for the closed-loop system.
- We also can strengthen the stability for linear systems from asymptotic stability to exponential stability based on the form of the Lyapunov function.

Enlarging the class of systems

- The LQR convergence result in Lemma 1 is the simplest to establish, but we can enlarge the class of systems and penalties for which closed-loop stability is guaranteed.
- The system restriction can be weakened from controllability to *stabilizability*, which is discussed in Exercises 1.19 and 1.20.
- The restriction on the allowable state penalty Q can be weakened from $Q > 0$ to $Q \geq 0$ and (A, Q) *detectable*, which is also discussed in Exercise 1.20.
- The restriction $R > 0$ is retained to ensure uniqueness of the control law.
- In applications, if one cares little about the cost of the control, then R is chosen to be small, but positive definite.

Constrained regulation

- Now we add the constraints to the control problem.
- Control Objective.

$$V(x(0), \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{N-1} \left[x(k)' Q x(k) + u(k)' R u(k) \right] + \frac{1}{2} x(N)' P_f x(N)$$

- Constraints.

$$Fx(k) + Eu(k) \leq e, \qquad k = 0, 1, \ldots N-1$$

- Optimization.

$$\min_{\mathbf{u}} V(x, \mathbf{u})$$

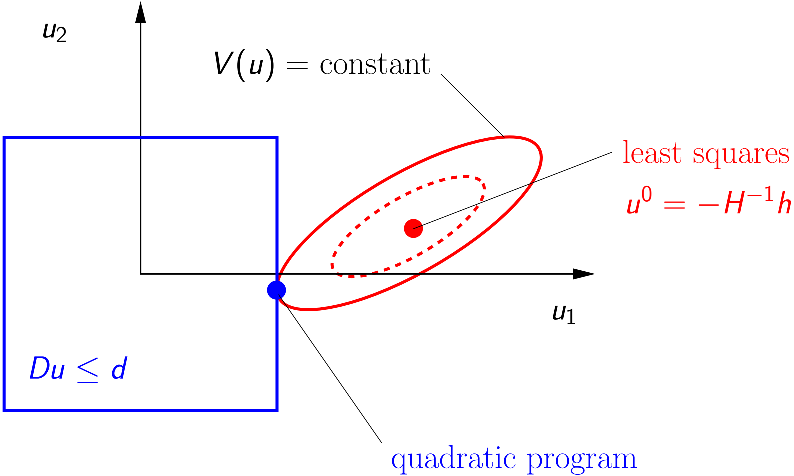
subject to the model and constraints.

Adding constraints is a big change!

- We cannot solve the constrained problem in closed-form using dynamic programming.
- Now we have a quadratic objective subject to linear constraints, which is known as a quadratic program (Nocedal and Wright, 1999).
- We must compute the solution for industrial-sized multivariable problems *online*.
- We must compute all the $u(k), k = 1, 2, \dots, N - 1$ simultaneously. That means a bigger online computation.
- But we have fast, reliable online computing available; why not use it?

The geometry of quadratic programming

$$\min_u V(u) = (1/2)u'Hu + h'u \qquad \text{subject to } Du \leq d$$



The simplest possible constrained control law

- The optimal control law for the regulation problem with an active input constraint remains linear (affine) $u = Kx + b$.
- But the control laws are valid over only local polyhedral regions
- The simplest example would be a SISO, first-order system with input saturation. There are three regions.

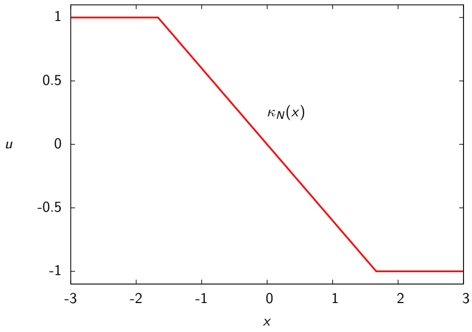


Figure 9: The optimal control law for $x^+ = x + u$, $N = 2$, $Q = R = 1$, $u \in [-1, 1]$.

Stability of the closed-loop system

- We denote the solution to the QP, which depends on the initial state x as
$$u^0(x) = \kappa_N(x)$$
- The constrained controller is now nonlinear even though the model is linear
- The closed-loop system is then
$$x^+ = Ax + B\kappa_N(x)$$
- And we would like to design the controller (choose N, Q, R, P_f) so that the closed-loop system is stable.

Terminal constraint solution

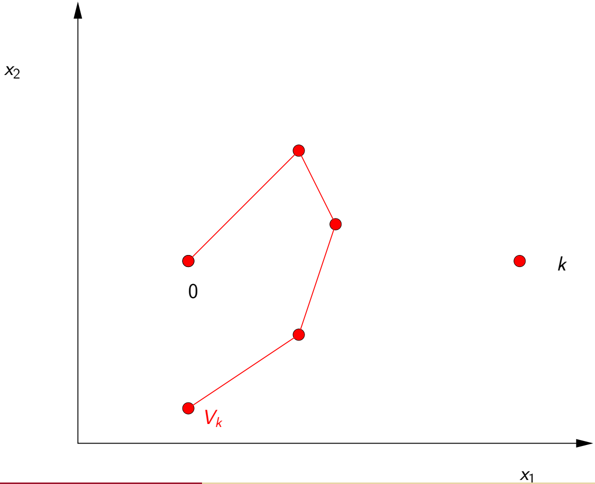
- Adding a terminal constraint $x(N) = 0$ ensures stability

Terminal constraint solution

- Adding a terminal constraint $x(N) = 0$ ensures stability
- May cause infeasibility (cannot reach 0 in only N moves)

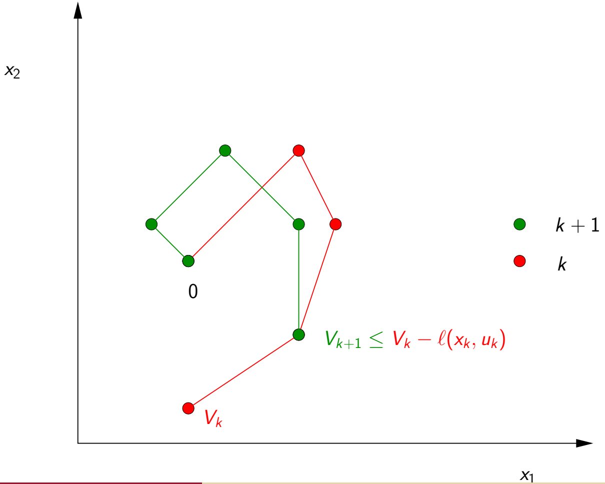
Terminal constraint solution

- Adding a terminal constraint $x(N) = 0$ ensures stability
- May cause infeasibility (cannot reach 0 in only N moves)
- Open-loop predictions not equal to closed-loop behavior



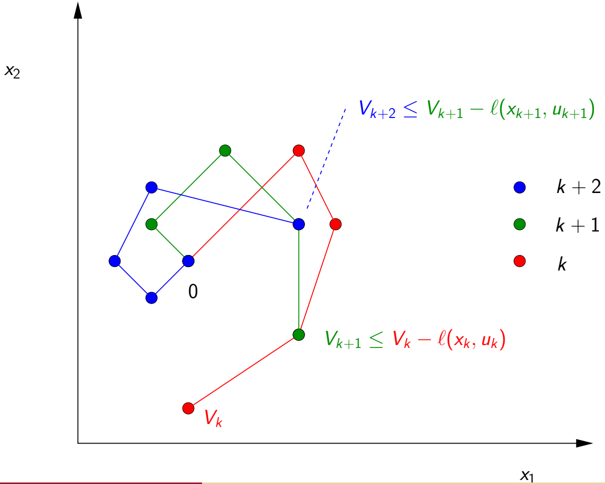
Terminal constraint solution

- Adding a terminal constraint $x(N) = 0$ ensures stability
- May cause infeasibility (cannot reach 0 in only N moves)
- Open-loop predictions not equal to closed-loop behavior



Terminal constraint solution

- Adding a terminal constraint $x(N) = 0$ ensures stability
- May cause infeasibility (cannot reach 0 in only N moves)
- Open-loop predictions not equal to closed-loop behavior



Infinite horizon solution

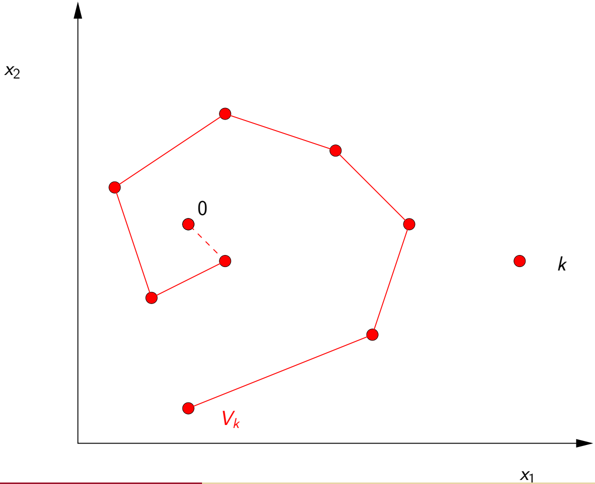
- The infinite horizon ensures stability

Infinite horizon solution

- The infinite horizon ensures stability
- Open-loop predictions equal to closed-loop behavior (desirable!)

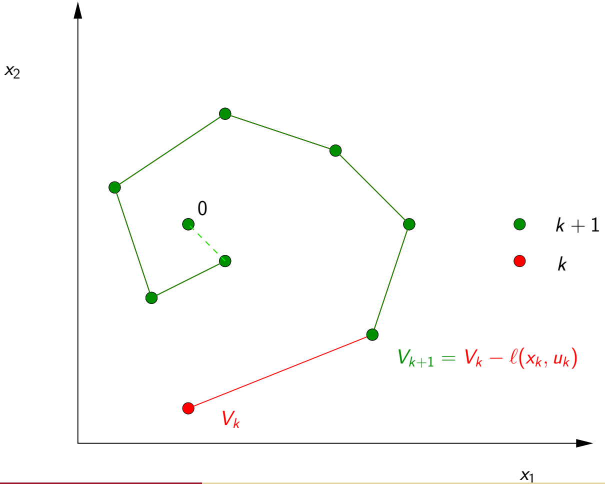
Infinite horizon solution

- The infinite horizon ensures stability
- Open-loop predictions equal to closed-loop behavior (desirable!)
- Challenge to implement



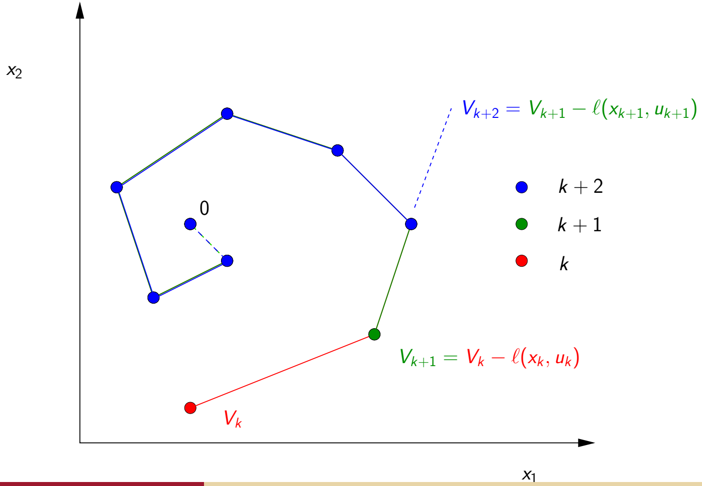
Infinite horizon solution

- The infinite horizon ensures stability
- Open-loop predictions equal to closed-loop behavior (desirable!)
- Challenge to implement



Infinite horizon solution

- The infinite horizon ensures stability
- Open-loop predictions equal to closed-loop behavior (desirable!)
- Challenge to implement



Implementation challenges

- The theory is now in reasonably good shape. In 1990, there was no theory addressing the control problem with constraints.
- Numerical QP solution methods are efficient and robust.
- In the process industries, applications with hundreds of states and inputs are running today.
- But challenges remain:
  - Scale. Methods must scale well with horizon N, number of states n, and inputs, m. The size of applications continues to grow.
  - Ill-conditioning. Detect and repair nearly collinear constraints, badly conditioned Hessian matrix.
  - Model accuracy. How to adjust the models automatically as conditions in the plant change.

State estimation

- In most applications, the variables that are conveniently or economically measurable (y) are a small subset of the state variables required to model the system (x).
- Moreover, the measurement is corrupted with sensor noise and the state evolution is corrupted with process noise.
- Determining a good state estimate for use in the regulator in the face of a noisy and incomplete output measurement is a challenging task.
- This is the challenge of state estimation.

Noise in the data

- To fully appreciate the fundamentals of state estimation, we must address the fluctuations in the data.

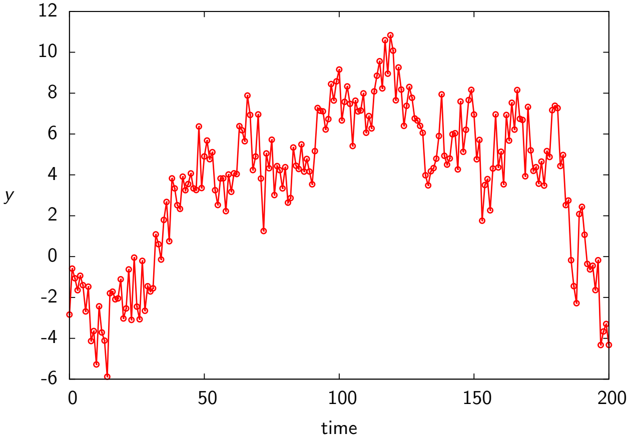


Figure 10: A measurement with noise

Deterministic and stochastic systems

- The motivation for stochastic models is to account for the random effects of the environment (disturbances) on the system under study.
- Some of the observed fluctuation in the data is assignable to the measurement device. This source of fluctuation is known as measurement "noise."
- Some of the observed fluctuation in the data is assignable to unmodeled disturbances from the environment affecting the state of the system.
- The simplest stochastic model for representing these two possible sources of disturbances is a linear model with added random variables

$$\begin{aligned} x^+ &= Ax + Bu + w \\ y &= Cx + Du + v \end{aligned}$$

with initial condition $x(0) = x_0$.

Kalman filter

- The random variable v (p vector) is used to model the measurement noise and w (n vector) models the random part of the process disturbance.
- Given a *linear* discrete time model subject to *normally distributed* process and measurement noise, the optimal state estimator is known as the Kalman filter (Kalman, 1960a).
- Rather than take this up this stochastic version of the problem, we will first derive an optimal deterministic least-squares estimator.
- It can be shown that this deterministic least-squares estimator and the optimal stochastic (Kalman) filter are equivalent.

- We first consider the state estimation problem as a deterministic optimization problem rather than an exercise in maximizing conditional density.
- Consider a time horizon with measurements $y(k), k = 0, 1, \dots, T$. We consider the prior information to be our best initial guess of the initial state $x(0)$, denoted $\overline{x}(0)$, and weighting matrices $P^{-}(0)$, Q, and R for the initial state, process disturbance, and measurement disturbance.
- A reasonably flexible choice for objective function is

$$V_T(\mathbf{x}(T)) = \frac{1}{2}\Bigg(|x(0) - \overline{x}(0)|^2_{(P^{-}(0))^{-1}} + \sum_{k=0}^{T-1} |x(k+1) - Ax(k)|^2_{Q^{-1}} + \sum_{k=0}^{T} |y(k) - Cx(k)|^2_{R^{-1}} \Bigg) \quad (14)$$

in which $\mathbf{x}(T) := \{x(0), x(1), \dots, x(T)\}$.

Least-squares estimation

- We claim that the following (deterministic) least squares optimization problem produces the same result as the conditional density function maximization of the Kalman filter

$$\min_{\mathbf{x}(T)} V_T(\mathbf{x}(T)) \tag{15}$$

- See, for example, (Rawlings and Mayne, 2009, pp. 28–32) for a proof of this claim.

Let's first examine a scalar system with $T = 1$

- We have only one state to optimize, x_0.
- The objective is

$$V(x_0) = (1/2)\left(\frac{(x_0 - \overline{x}_0)^2}{p_0^-} + \frac{(y_0 - cx_0)^2}{r}\right)$$

- Take derivative dV/dx_0, set to zero

$$\begin{aligned}
\frac{d}{dx_0} V &= \frac{(x_0 - \overline{x}_0)}{p_0^-} + \frac{-c(y_0 - cx_0)}{r} \\
0 &= \frac{(x_0 - \overline{x}_0)}{p_0^-} + \frac{-c(y_0 - cx_0)}{r} \\
0 &= r(x_0 - \overline{x}_0) - p_0^- c(y_0 - cx_0) \\
0 &= (c^2 p_0^- + r)x_0 + -r\overline{x}_0 - p_0^- cy_0
\end{aligned}$$

Scalar system with $T = 1$

- Rearrange and solve for optimal estimate, denoted $\hat{x}_0$

$$\hat{x}_0 = \frac{r}{c^2 p_0^- + r}\overline{x}_0 + \frac{p_0^- c}{c^2 p_0^- + r} y_0$$

$$\hat{x}_0 = \frac{c^2 p_0^- + r}{c^2 p_0^- + r}\overline{x}_0 + \frac{p_0^- c}{c^2 p_0^- + r}(y_0 - c\overline{x}_0)$$

$$\hat{x}_0 = \overline{x}_0 + \frac{p_0^- c}{c^2 p_0^- + r}(y_0 - c\overline{x}_0)$$

- As in regulation, we see a linear update formula, with estimator gain ℓ_0 operating on our prior fitting error $y_0 - c\overline{x}_0$.

$$\hat{x}_0 = \underbrace{\overline{x}_0}_{\text{prior}} + \underbrace{\ell_0}_{\text{gain}} \underbrace{(y_0 - c\overline{x}_0)}_{\text{fitting error}} \qquad \ell_0 = \frac{p_0^- c}{c^2 p_0^- + r}$$

Back to vectors and matrices ($T = 1$)

- Quadratic functions in x_0

$$V(x_0) = (1/2)\left((x_0 - \overline{x}_0)'(P_0^-)^{-1}(x_0 - \overline{x}_0) + (y_0 - Cx_0)'R^{-1}(y_0 - Cx_0)\right)$$

- Take derivative, set to zero, solve[3]

$$\hat{x}_0 = \overline{x}_0 + L_0(y_0 - C\overline{x}_0) \qquad L_0 = P_0^- C'(CP_0^- C' + R)^{-1}$$

- Note the similarity to the scalar case
- The cost function can then be written as

$$V(x_0) = (1/2)(x_0 - \hat{x}_0)'P_0^{-1}(x_0 - \hat{x}_0) + \text{constant}$$
$$P_0 = P_0^- - P_0^- C'(CP_0^- C' + R)^{-1} CP_0^-$$

[3] Note that you require the matrix inversion lemma for this step in the vector/matrix case. See (Rawlings and Mayne, 2009, p.34) for details.

Adding subsequent measurements; $T > 1$

- Just as in regulation, we can use (forward) dynamic programming to optimize over each x_k as new measurements arrive.
- The result is a two-step procedure.
- Adding the measurement at time k produces

$$\begin{aligned}
\hat{x}(k) &= \hat{x}^-(k) + L(k)(y(k) - C\hat{x}^-(k)) \\
L(k) &= P^-(k)C'(CP^-(k)C' + R)^{-1} \\
P(k) &= P^-(k) - P^-(k)C'(CP^-(k)C' + R)^{-1}CP^-(k)
\end{aligned}$$

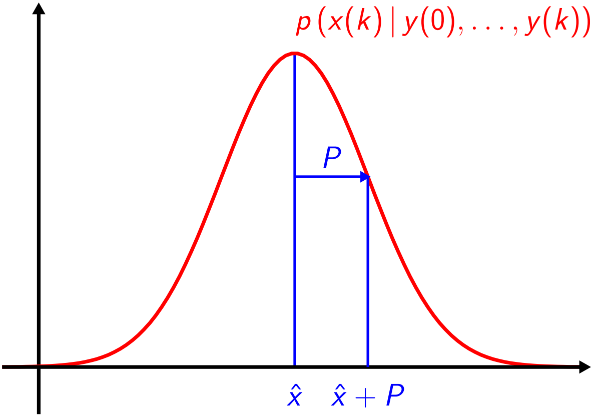
- Propagating the model to time $k+1$

$$\begin{aligned}
\hat{x}^-(k+1) &= A\hat{x}(k) \\
P^-(k+1) &= AP(k)A' + Q
\end{aligned}$$

- Notice that the recursion provides a highly efficient online procedure for estimating the state.

The stochastic interpretation of $\hat{x}$ and P

$\hat{x}$ and P are estimates of parameters in the conditional probability of $x(k)$ given $y(0), \ldots, y(k)$.



Optimal stochastic filter and recursive least squares

- The discovery (and rediscovery) of the close connection between recursive least squares and optimal statistical estimation has not always been greeted happily by researchers:

  > *The recursive least squares approach was actually inspired by probabilistic results that automatically produce an equation of evolution for the estimate (the conditional mean). In fact, much of the recent least squares work did nothing more than rederive the probabilistic results (perhaps in an attempt to understand them). As a result, much of the least squares work contributes very little to estimation theory.*
  > *— Jazwinski (1970, pp.152–153)*

Optimal stochastic filter and recursive least squares

- In contrast with this view, we find both approaches valuable.
- The probabilistic approach, which views the state estimator as maximizing conditional density of the state given measurement, offers the most insight. It provides a rigorous basis for comparing different estimators based on the variance of their estimate error.
- It also specifies what information is required to define an optimal estimator, with variances Q and R of primary importance.
- In the probabilistic framework, these parameters should be found from modeling and data.
- The main deficiency in the least squares viewpoint is that the objective function, although reasonable, is ad hoc and not justified.
- The choice of weighting matrices Q and R is arbitrary.
- If we restrict attention to unconstrained linear systems, the probabilistic viewpoint is clearly superior.

Optimal stochastic filter and recursive least squares

- Approaching state estimation with the perspective of least squares pays off, however, when the models are significantly more complex. It is generally intractable to find and maximize the conditional density of the state given measurements for complex, nonlinear and constrained models.
- Reasonable and useful objective functions can be chosen for even complex, nonlinear and constrained models.
- Moreover, knowing which least squares problems correspond to which statistically optimal estimation problems for the simple linear case, provides the engineer with valuable insight in choosing useful objective functions for nonlinear estimation.

Let's try out the Kalman filter

Example 2

Let's estimate a "constant" from noisy measurements. Say we want to know the temperature in this room. We don't expect it to be changing on a short time scale, and we have a quite noisy temperature measurement. Since $x(k)$ is supposed to be constant, and we measure it, the model is very simple

$$x^+ = x + w$$
$$y = x + v$$

in which $A = I, C = I$, and we assume w and v are zero mean random variables with variances Q and R. We expect that $R \gg Q$.
We make an initial guess $x_0 = 70^\circ\mathrm{F}$ and assign some uncertainty to that guess, $P_0 = 1$. Let the true temperature be $x = 75^\circ\mathrm{F}$. Examine the performance of the Kalman filter in estimating the temperature as measurements become available.

Implementing the KF is easy

Solution

```
nsim = 500; time = [0:nsim-1];
x0 = 75; xhat0 = 70;
P0 = 1;
Q = 0; R = 1;
x(1) = x0; xhat_(1) = xhat0;
P_(1) = P0;

for k = 1:nsim
  v = sqrt(R)*randn;
  y(k) = x(k) + v;
  L(k) = P_(k)/(P_(k)+R);
  xhat(k) = xhat_(k) + L(k)*(y(k)-xhat_(k));
  P(k) = P_(k) - P_(k)/(P_(k)+R)*P_(k);
  if (k == nsim) break endif
    w = sqrt(Q)*randn;
    x(k+1) = x(k) + w;
    xhat_(k+1) = xhat(k);
    P_(k+1) = P(k) + Q;
endfor

plot(time,xhat, time, y, 'x')
```

Solution

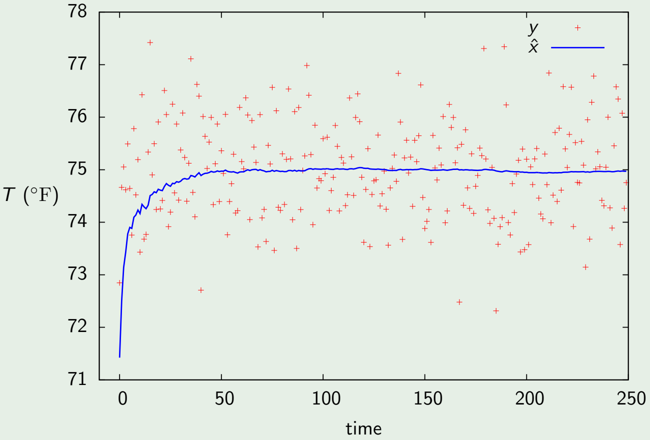


Figure 11: Noisy measurement and state estimate versus time when estimating a scalar constant.

How about just taking the sample mean for the estimate?

Solution

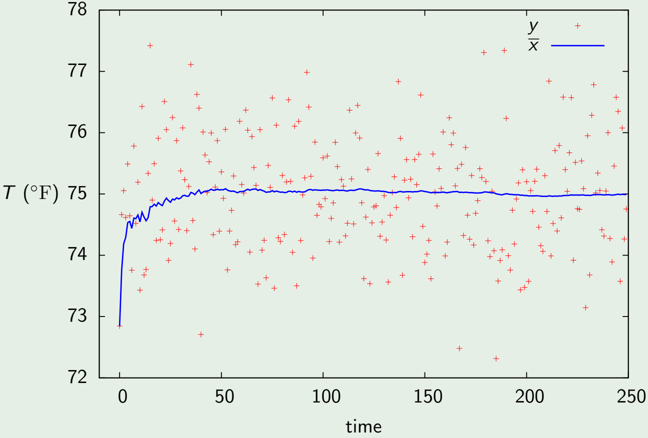


Figure 12: Taking the sample mean as the estimate

No process noise means we're eventually certain!

Solution

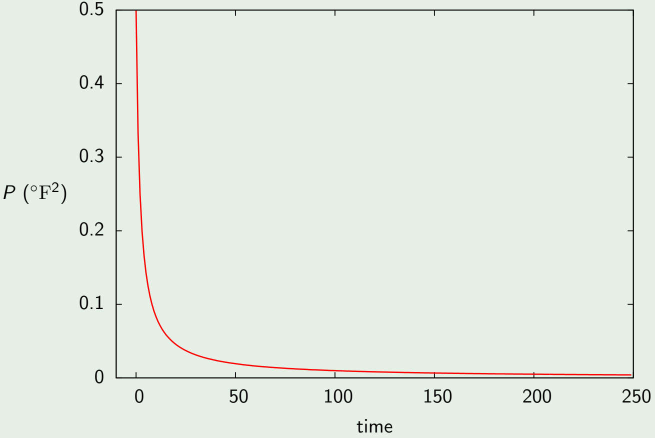


Figure 13: Estimate error variance goes to zero without process noise

Now let's add some process noise

Choose $Q > 0$

$$x^+ = x + w$$
$$y = x + v$$

w and v are zero mean random variables with variances $Q = 0.1$ and $R = 1$. In simulation, we use

```
w = sqrt(Q)*randn;
v = sqrt(R)*randn;
```

Process noise leads to nonzero steady-state variance

Solution

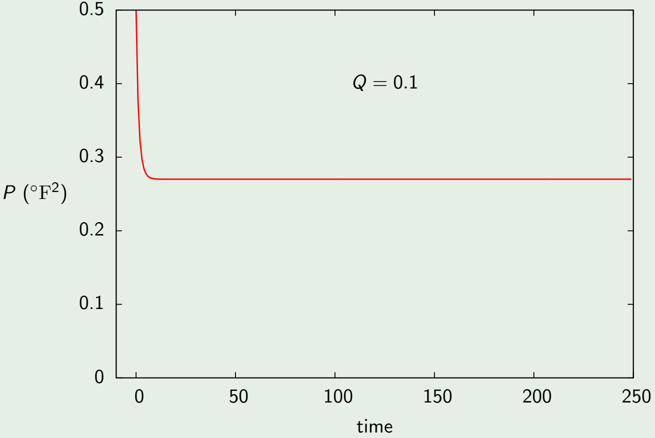


Figure 14: Process noise leads to nonzero steady-state estimate error variance

Notice now that x is on the move

Solution

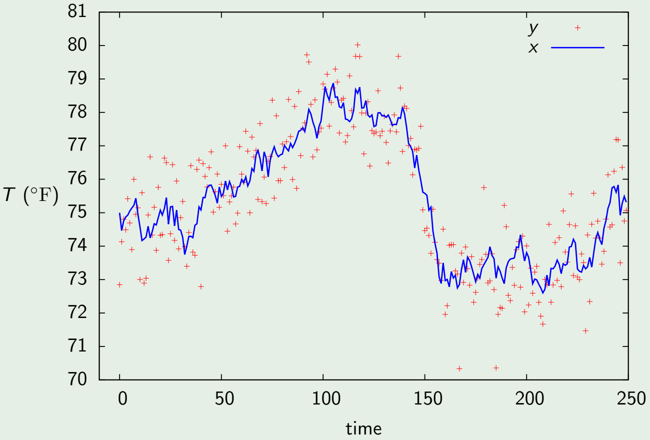


Figure 15: With process noise, the state x is moving

And our estimate $\hat{x}$ is on the chase

Solution

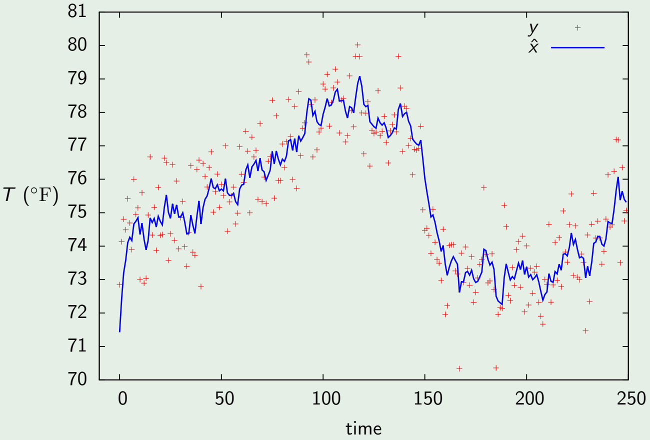


Figure 16: The optimal estimate $\hat{x}$ is chasing after x

And the sample mean is no longer a good estimator

Solution

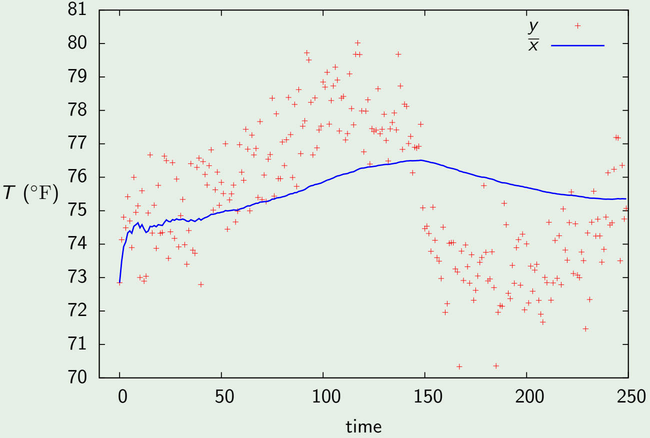


Figure 17: The sample mean is no longer a good estimator

Observability

- We next explore the convergence properties of the state estimators. For this we require the concept of system observability.
- The basic idea of observability is that any two distinct states can be *distinguished* by applying some input and observing the two system outputs over some finite time interval (Sontag, 1998, p.262–263).
- This general definition is discussed in more detail when treating nonlinear systems in Rawlings and Mayne (2009, Chapter 4), but observability for linear systems is much simpler.
- First of all, the applied input is irrelevant and we can set it to zero. Therefore consider the linear time-invariant system (A, C) with zero input

$$\begin{aligned} x(k+1) &= Ax(k) \\ y(k) &= Cx(k) \end{aligned}$$

Observability

- The system is observable if there exists a finite N, such that for every $x(0)$, N measurements $\{y(0), y(1), \ldots, y(N-1)\}$ distinguish uniquely the initial state $x(0)$.
- Similarly to the case of controllability, if we cannot determine the initial state using n measurements, we cannot determine it using $N > n$ measurements.
- Therefore we can develop a convenient test for observability as follows. For n measurements, the system model gives

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0) \tag{16}$$

Observability test

- The question of *observability* is therefore a question of *uniqueness* of solutions to these linear equations. The matrix appearing in this equation is known as the *observability matrix* $\mathcal{O}$

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{17}$$

- From the fundamental theorem of linear algebra, we know the solution to (16) is unique if and only if the *columns* of the $np \times n$ observability matrix are linearly independent.
- Therefore, we have that the system (A, C) is observable if and only if

$$\mathrm{rank}(\mathcal{O}) = n$$

Convergence of the state estimator

- Next we consider the question of convergence of the estimates of the optimal estimators we have considered.
- The simplest convergence question to ask is the following. Given an initial estimate error, and zero state and measurement noises, does the state estimate converge to the state as time increases and more measurements become available?
- If the answer to this question is yes, we say the estimates converge; sometimes we say the estimator converges.

Optimality of estimator does not ensure convergence

- As with the regulator, optimality of an estimator does not ensure its stability. Consider the case $A = I$, $C = 0$. The optimal estimate is $\hat{x}(k) = \overline{x}(0)$, which does not converge to the true state unless we have luckily chosen $\overline{x}(0) = x(0)$.
- If we could count on that kind of luck, we would have no need for state estimation!
- Obviously the lack of stability is caused by our choosing an unobservable (undetectable) system.

Combining the two steps of the Kalman filter

- We can combine the two steps of the Kalman filter into one step. The estimate is

$$
\begin{aligned}
\hat{x}^-(k+1) &= A\hat{x}(k) \\
&= A\Big(\hat{x}^-(k) + L(y(k) - C\hat{x}^-(k))\Big) \\
\hat{x}^-(k+1) &= A\hat{x}^-(k) + \tilde{L}(y(k) - C\hat{x}^-(k))
\end{aligned}
$$

  with $\tilde{L} = AL$.

- And the steady-state Riccati equation and estimator gain are

$$
\begin{aligned}
P^- &= Q + AP^-A' - AP^-C'(CP^-C' + R)^{-1}CP^-A' \\
\tilde{L} &= AP^-C'(CP^-C' + R)^{-1}
\end{aligned}
$$

Evolution of estimate error

- We define estimate error as

$$\tilde{x} = x - \hat{x}^-$$

- We then have

$$\begin{aligned}
\tilde{x}(k+1) &= x(k+1) - \hat{x}^-(k+1) \\
&= Ax(k) - \left(A\hat{x}^-(k) + \tilde{L}(y(k) - C\hat{x}^-(k))\right) \\
&= A\big(x(k) - \hat{x}^-(k)\big) - \tilde{L}\big(Cx(k) - C\hat{x}^-(k)\big) \\
\tilde{x}(k+1) &= (A - \tilde{L}C)\tilde{x}(k)
\end{aligned}$$

Of what does this remind us?

Estimator

$$\begin{aligned}\tilde{x}^+ &= (A - \tilde{L}C)\tilde{x} \\ \tilde{L} &= AP^-C'(CP^-C' + R)^{-1} \\ P^- &= Q + AP^-A' - AP^-C'(CP^-C' + R)^{-1}CP^-A'\end{aligned}$$

Regulator!

$$\begin{aligned}x^+ &= (A + BK)x \\ K &= -(B'\Pi B + R)^{-1}B'\Pi A \\ \Pi &= Q + A'\Pi A - A'\Pi B(B'\Pi B + R)^{-1}B'\Pi A\end{aligned}$$

Duality of regulation and estimation

Duality variables

Regulator	Estimator
A	A'
B	C'
Π	P^-
Q	Q
R	R
K	$-\tilde{L}'$
$A + BK$	$(A - \tilde{L}C)'$
x	$\tilde{x}'$

Regulator	Estimator
$R > 0, \quad Q > 0$	$R > 0, \quad Q > 0$
(A, B) controllable	(A, C) observable

Two for the price of one

Establishing stability of the regulator $A + BK$ establishes stability of the estimator!

Estimator convergence

Lemma 3 (Estimator convergence)

For (A, C) *observable,* $Q, R > 0$*, and noise-free measurements* $\mathbf{y}(k) = \left\{ Cx(0), CAx(0), \ldots, CA^k x(0) \right\}$*, the optimal linear state estimate converges to the state (estimate error converges to zero)*

$$\tilde{x}(k) \to 0 \quad \textit{as } k \to \infty$$

Enlarging the class of systems

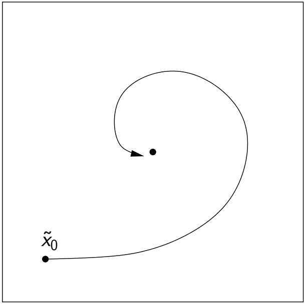
- The estimator convergence result in Lemma 3 is the simplest to establish, but, as in the case of the LQ regulator, we can enlarge the class of systems and weighting matrices (variances) for which estimator convergence is guaranteed.
- The system restriction can be weakened from observability to *detectability*, which is discussed in Exercises 1.31 and 1.32.
- The restriction on the process disturbance weight (variance) Q can be weakened from $Q > 0$ to $Q \geq 0$ and (A, Q) *stabilizable*, which is discussed in Exercise 1.33.
- The restriction $R > 0$ remains to ensure uniqueness of the estimator.

But we have noise in the system!

Now we keep the noise, $x^+ = Ax + w, \quad y = Cx + v$, and estimate error satisfies

$$
\begin{aligned}
\tilde{x}(k+1) &= x(k+1) - \hat{x}^-(k+1) \\
&= Ax(k) + w - \left(A\hat{x}^-(k) + \tilde{L}(y(k) - C\hat{x}^-(k)) \right) \\
&= A\big(x(k) - \hat{x}^-(k)\big) + w - \tilde{L}\big(Cx(k) + v - C\hat{x}^-(k)\big) \\
\tilde{x}(k+1) &= (A - \tilde{L}C)\tilde{x}(k) + w - \tilde{L}v
\end{aligned}
$$

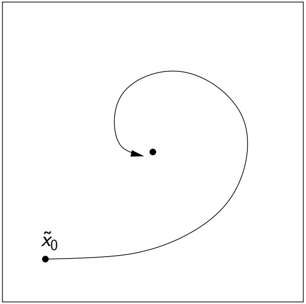
Behavior with and without disturbances

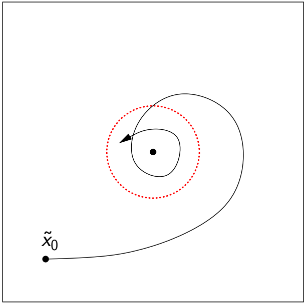


Nominal System

$$\tilde{x}^+ = (A - \tilde{L}C)\tilde{x}$$

Behavior with and without disturbances





Nominal System

$$\tilde{x}^+ = (A - \tilde{L}C)\tilde{x}$$

System with Disturbance

$$\tilde{x}(k+1) = (A - \tilde{L}C)\tilde{x}(k) + w - \tilde{L}v$$

w is the process disturbance
v is the measurement disturbance

Further reading I

R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

D. P. Bertsekas. *Dynamic Programming*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Engineering*, pages 35–45, March 1960a.

R. E. Kalman. Contributions to the theory of optimal control. *Bull. Soc. Math. Mex.*, 5: 102–119, 1960b.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009. 576 pages, ISBN 978-0-9759377-0-9.

E. D. Sontag. *Mathematical Control Theory*. Springer-Verlag, New York, second edition, 1998.

G. Strang. *Linear Algebra and its Applications*. Academic Press, New York, second edition, 1980.

Review

Lab Exercises — LQR

- Exercise 1.25
- Exercise 2.3
- Exercise 2.19
- Exercise 6.2

Lab Exercises — LQE

- Exercise 1.30
- Exercise 1.31
- Exercise 1.34
- Exercise 1.36
- Exercise 1.41
- Exercise 1.43