CSE599i: Online and Adaptive Machine Learning **Winter 2018**

Lecture 20: Linear Dynamics and LQG

Lecturer: Kevin Jamieson *Scribes: Atinuke Ademola-Idowu, Yuanyuan Shi*

1 Introduction

As an introduction to linear system dynamics and controls, we begin with a motivating example. Consider a simple helicopter dynamics where the objective is to move the helicopter from current position h to hover about a new height h_D (see Fig. 1).

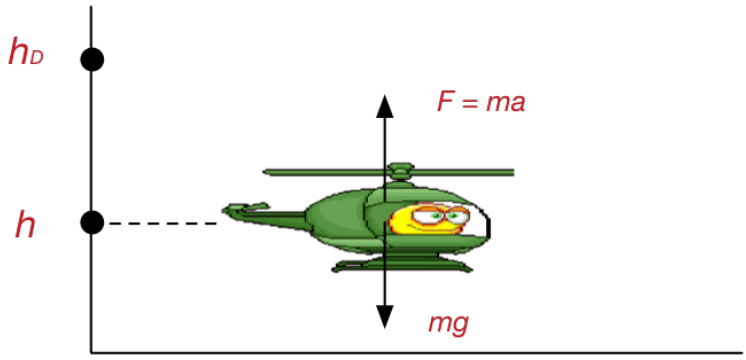

Figure 1: Helicopter motion dynamics

The dynamics for this case is governed by the equation of motion, which in discrete form is:

$$\begin{aligned} h_{t+1} &= h_t + \Delta v_t + \frac{1}{2}\Delta^2(a_t - g) \\ v_{t+1} &= v_t + \Delta(a_t - g) \end{aligned} \tag{1}$$

where h_t is the position in meters (m), v_t is the velocity in (m/s), a_t is the acceleration in (m/s^2), g is the gravity also in (m/s^2), and Δ is the time step in (s), all at time t.

The helicopter dynamics in (1) is called a linear system because it depends linearly on the state variables. It can be re-written in a form called the state-space as:

$$\underbrace{\begin{bmatrix} h_{t+1} \\ v_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta^2 \\ \Delta \end{bmatrix}}_{B} \underbrace{(a_t - g)}_{u_t}. \tag{2}$$

In general, the dynamics of a discrete linear time-invariant (LTI) system can be represented in a state-space form [1]:

$$x_{t+1} = Ax_t + Bu_t \tag{3}$$

where x_t is the system state at time t and it is considered to be the smallest subset of system variables from which other system variables can be obtained. u_t is the control input at time t which can be used in steering the system to a desired state. The matrix A is the system matrix which governs the evolution of the states from one time step to another and the matrix B is the control matrix which determines how the system input affects the states. Equation (3) is known as the system state equation.

Going back to our example, the states of the helicopter dynamics can therefore be chosen as the position h and the velocity v because the motion of the helicopter can be completely described in terms of these two variables.

In most cases, simply achieving the desired objective of moving the helicopter to a desired height h_D is not sufficient. We often want to achieve this objective at the lowest cost possible while keeping the system stable, where the cost in this case could be time, fuel, effort and so on. We therefore want to find the best (optimal) control action/input u_t that can achieve our stated goal at minimum cost. This requirement can be captured mathematically as finding a control action sequence $u_t,\ t = 1, 2, ..., T$, that minimizes over a time horizon T, the following objective function:

$$J(u) = \sum_{t=1}^{T} x_t^T Q x_t + u_t R u_t. \tag{4}$$

For ease of computation of our example problem, the state of the system is redefined as the deviation from the desired state, that is $x_t := \left(\begin{bmatrix} h_t \\ v_t \end{bmatrix} - x_D \right)$ where the desired state $x_D := \begin{bmatrix} h_D \\ 0 \end{bmatrix}$. The control action u_t is the force to be applied and has to be greater than gravity g for the helicopter to move. The matrix Q and R in (4) are positive definite matrices which places weights on the competing objectives in (4), (that is, $x_t^T Q x_t$ which captures the deviation from the desired state and $u_t R u_t$ which captures the control effort to be used), based on which of the objectives we care more about.

Representing the helicopter dynamics in the form (2) and its performance objective in the form (4) allows us to make an interesting connection with the Markov Decision Process (MDP) from previous lectures. If we let the control input u_t at time t be drawn from a feasible action space, then the state equation in (3) is a special case of the MDP. This is true because the state transition matrix in MDP gives the probabilities of moving from one state to another. In this case, since we know the system dynamics, it becomes deterministic which means that

$$p(x, u, y) = p(x_{t+1} = y | x_t = x, u_t = u) = 1 \tag{5}$$

This implies that the theories developed in previous lectures for the probabilistic MDP cases can also be extended to this deterministic case by replacing the state transition matrix with the system dynamics.

To find the sequence of control actions that minimizes the problem in (4) and also keeps the system stable, we consider the form the control action has to take. A discrete system such as in (3) is said to be stable if all the eigenvalues of the system matrix A are within the unit circle, that is, $\lambda_i \leq 1,\ i = 1, 2, \ldots, n$. The desired form of the control action is a feedback form $u_t = f(x_t)$, that is, to determine the next state of the system, the control action will be a function of the current state of the system. The optimal control action to be taken can be obtained by solving the Linear Quadratic Regulator (LQR) problem or the Linear Quadratic Gaussian (LQG) problem which are introduced in the next section.

2 Linear System Optimal Control

2.1 Linear quadratic regulator (LQR): Discrete-time finite horizon

As we have briefly talked about at the end of last section, the helicopter example actually represents a large class of control problems which calls linear quadratic regulator (LQR), where the goal is to find the optimal sequence of control that minimizes a quadratic cost function subject to the linear system dynamics.

Suppose the time horizon is finite and we define the system state x_t with reference to x_D, we have the following optimization problem called "finite horizon LQR",

$$\min_{u_0,...,u_{N-1}} J(u_0, ..., u_{N-1}, x_0) = \sum_{t=0}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + x_N^T Q_f x_N \tag{6a}$$

$$s.t. \;\; x_{t+1} = Ax_t + Bu_t, t = 0, 1, ..., N-1 \tag{6b}$$

where $Q, R, Q_f \geq 0$ are positive semi-definite matrices.

- N is called time horizon (we will consider $N = \infty$ later);
- $x_t^T Q x_t$ measures the state deviation cost at time t;
- $u_t^T R u_t$ measures control input cost at time t;
- $x_N^T Q_f x_N$ measures the terminal state deviation cost.

In order to solve the above optimization problem in (6), we use dynamic programming method. The basic idea of dynamic programming is that the optimal control sequence over the entire horizon remains optimal at intermediate points in time. To begin this discussion, we will embed the optimization problem which we are solving in (6) to a larger class of problems. More specifically, we will consider the original cost function of equation (6a) from an arbitrary initial time t. Then the cost function from time t to N is written as,

$$J(u_t, ..., u_N, x_t) = \sum_{\tau=t}^{N-1} (x_\tau^T Q x_\tau + u_\tau^T R u_\tau) + x_N^T Q_f x_N \,, \tag{7}$$

Bellmans principle of optimality says that *if we have found the optimal trajectory on the interval from* $\{0, 1, ..., N\}$ *by solving the optimal control problem in* (6)*, the resulting trajectory is also optimal on all subintervals of this interval of the form* $\{t, t+1, ..., N\}$ *with* $t > 0$*, provided that the initial condition* x_t *at time t was obtained from running the system forward along the optimal trajectory from time 0.* This statement is obvious since if the optimal overall trajectory is suboptimal for certain subinterval $\{t, t+1, ..., N\}$, we could replace that subinterval trajectory with a better one to achieve better overall performance, which contradicts the overall optimality assumption. Therefore, the optimal trajectory on the interval from $\{0, 1, ..., N\}$ is also optimal on all subintervals $\{t, t+1, ..., N\}$ with $t > 0$.

Define the optimal value of $J(u_t, ..., u_N, x_t)$ starting from state x_t as $V^*(x_t)$, where $V^*(x_t)$ is called the optimal "cost-to-go" (or optimal value function) at state x_t. The LQR objective function in (6a) is thus equivalently to find $V^*(x_0)$ since,

$$V^*(x_0) = \min_{u_0,...,u_{N-1},x_0} J(u_0, ..., u_{N-1}, x_0) \,,$$

Now suppose at time t, we know $V^*(x_{t+1})$ for all possible next step states x_{t+1} and we are interested in finding the optimal u_t (and hence $V^*(x_t)$). Bellman's principle of optimality suggests,

$$u_t^* = \arg\min_{u_t} \left(\underbrace{x_t^T Q x_t + u_t^T R u_t}_{\text{instant reward at time t}} + \underbrace{V^*(Ax_t + Bu_t)}_{\text{next state cost-to-go}} \right) , \tag{8}$$

and the optimal cost-to-go at time t is,

$$V^*(x_t) = \left(x_t^T Q x_t + {u_t^*}^T R u_t^* + V^*(Ax_t + Bu_t) \right) , \tag{9}$$

Therefore, we can find the optimal control sequence by solving the above equations (8) and (9) *backwards* starting at $t = N$, which leads to the following theorem.

Theorem 1. *The optimal cost-to-go and the optimal control at time t are given by:*

$$V^*(x_t) = x_t^T P_t x_t \, , \tag{10}$$

$$u_t^* = -K_t x_t \, , \tag{11}$$

where $t \in \{0, 1, ..., N-1\}$ *and,*

$$P_t = Q + K_t^T R K_t + (A - BK_t)^T P_{t+1} (A - BK_t), \quad P_N = Q_f \, , \tag{12}$$

$$K_t = (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \, , \tag{13}$$

Proof. We will prove the theorem by induction. For the final time step $i = N$, the optimal cost-to-go is trivially given by $V^*(x_N) = x_N^T Q_f x_N$ from cost function definition in (7) since no action is performed at time N. Thus, $P_N = Q_f$.

Now we assume the theorem hold for time $i = t$; our goal is to prove it holds for $i = t - 1$.

From Eq (9), we have

$$V^*(x_{t-1}) = \min_{u_{t-1}} [x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + V^*(Ax_{t-1} + Bu_{t-1})] \, ,$$

By induction hypothesis, $V^*(x_t) = x_t^T P_t x_t$. Therefore,

$$V^*(x_{t-1}) = \min_{u_{t-1}} (x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + (Ax_{t-1} + Bu_{t-1})^T P_t (Ax_{t-1} + Bu_{t-1})) \, , \tag{14}$$

The optimal u_{t-1} can be derived by setting the derivative of the above equation to zero:

$$\nabla_{u_{t-1}} V^*(x_{t-1}) = 2u_{t-1}^T R + 2(Ax_{t-1} + Bu_{t-1})^T P_t B = 0 \tag{15}$$

Hence the optimal control is,

$$u_{t-1}^* = -(R + B^T P_t B)^{-1} B^T P_t A x_{t-1} = -K_{t-1} x_{t-1} \tag{16}$$

To get the optimal value function, we can substitute optimal control in Eq. (16) in equation Eq.(14):

$$V^*(x_{t-1}) = x_{t-1}^T Q x_{t-1} + {u_{t-1}^*}^T R u_{t-1}^* + (Ax_{t-1} + Bu_{t-1}^*)^T P_t (Ax_{t-1} + Bu_{t-1}^*) \tag{17}$$

$$= x_{t-1}^T (Q + K_{t-1}^T R K_{t-1} + (A - BK_{t-1})^T P_t (A - BK_{t-1})) x_{t-1} \tag{18}$$

$$= x_{t-1}^T P_{t-1} x_{t-1} \tag{19}$$

Therefore, $P_{t-1} = Q + K_{t-1}^T R K_{t-1} + (A - BK_{t-1})^T P_t (A - BK_{t-1})$.

Eqs.(12) and (13) holds for time $i = t - 1$.

Therefore, theorem 1 has been proved by induction. □

Some insights:

Theorem 1 basically says that the optimal control law for linear system subject to quadratic cost is a *linear feedback controller*. At every time step, the optimal control input is a linear function of the current system state, where $u_t = -K_t x_t$. A demonstration of the optimal control law for linear system dynamics is provided in Fig. 2.

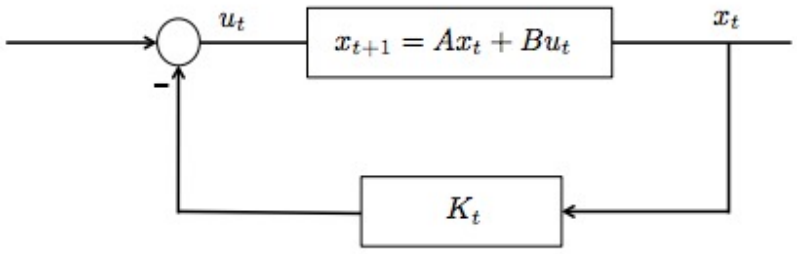

Figure 2: The optimal controller for LQR problem is a linear feedback controller $u_t = -K_t x_t$.

Therefore, the system dynamics under the optimal control law could be written as

$$x_{t+1} = Ax_t + Bu_t = Ax_t - BK_t x_t = (A - BK_t)x_t\,, \tag{20}$$

2.2 Infinite horizon LQR

If we consider the finite time horizon LQR problem in (6) in infinite time horizon, there is no so called "final step" so we need to drop out the last step cost term $x_N^T Q_f x_N$ in Eq. (6a).

Therefore we need to re-define the optimal cost-to-go function in $V^*(x)$ under the infinite time horizon case,

$$V^*(x_0) = \min_{u_0, u_1, \ldots} \sum_{t=0}^{\infty} (x_\tau^T Q x_\tau + u_\tau^T R u_\tau)\,, \tag{21}$$

subject to $x_{t+1} = Ax_t + Bu_t$

In finite horizon, we proved that the optimal cost-to-go $V^*(x_t) = x_t^T P_t x_t$ is a quadratic function of the current system state x_t, and P_t changes with respect to (w.r.t.) time t. Under the infinite horizon case, [1] shows that the optimal cost-to-go function $V^*(x)$ is still a quadratic form of state, where P is a time invariant matrix. To give a bit intuition about why P is constant: under the infinite horizon condition, all time steps look the "same" because the system evolution and cost function are the same. Therefore, instead of having a backward recursive formula of K_t and P_t, we have a stationary optimal policy K and optimal cost-to-go function.

From Bellman's optimality equation, for any state x

$$\begin{aligned} V^*(x) &= \min_u \{x^T Q x + u^T R u + V(Ax + Bu)\} \\ &= \min_u \{x^T Q x + u^T R u + (Ax + Bu)^T P (Ax + Bu)\} \end{aligned}$$

minimizing u is

$$u^* = -(R + B^T P B)^{-1} B^T P A x\,, \tag{22}$$

So the optimal value function

$$\begin{aligned} V^*(x) &= x^T Q x + u^{*T} R u^* + (Ax + Bu^*)^T P (Ax + Bu^*) \\ &= x^T (Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A) x \end{aligned}$$

this must hold for all x, so we conclude that P satisfies the Algebraic Riccati Equation (ARE):

$$P = (Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A)\,, \tag{23}$$

The solution of the ARE in (23) exists if the system described in the form of (3) is controllable. A system is said to be controllable if a control input can be applied to move the system states from an initial state to a final state in a finite time interval. If the system matrix $A \in \mathbb{R}^{n\times n}$ is full rank and its rank is n, then the system is controllable if the controllability matrix:

$$\mathcal{C} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \tag{24}$$

is also full rank, that is, n for $x \in R^n$ [7]. This guarantees that a controller can be found to achieve the stated objective. This definition follows from the observation that

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ &= A(Ax_{t-1} + Bu_{t-1}) + Bu_t \\ &= A^2x_{t-1} + ABu_{t-1} + Bu_t \\ &= A^3x_{t-2} + A^2Bu_{t-2} + ABu_{t-1} + Bu_t \end{aligned}$$

which makes it clear that it is sufficient for $\mathcal{C}$ to be full rank to generate an arbitrary state through some sequence of control input.

System controllability is roughly defined as the ability to do whatever we want with our system, or in more technical terms, the ability to transfer our system from any initial state $x(0) = x_0$ to any desired final state $x(N) = x_N$ in a finite time. Thus, the question to be answered is: can we find a control sequence $u(0), u(1), ..., u(N-1)$ such that $x(N) = x_N$?

For a linear system as,

$$x(k+1) = Ax(k) + Bu(k), x(0) = x_0\,,$$

we have the following set of equations,

$$x(1) = Ax(0) + Bu(0)\,, \tag{25}$$

$$x(2) = Ax(1) + Bu(1) = A(Ax(0) + Bu(0)) + Bu(1) \tag{26}$$

Once we solved P from (23), we plug back P to Eq.(22). The optimal control law in this case is given by a linear *constant* state feedback:

$$u^* = -Kx, \qquad K = (R + B^TPB)^{-1}B^TPA$$

The system dynamics under the infinite horizon LQR optimal controller could be written as

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ &= Ax_t - BKx_t \\ &= (A - BK)x_t \\ &= (A - BK)^2x_{t-1} \\ &= ... \\ &= (A - BK)^{t+1}x_0\,, \end{aligned}$$

Assuming the eigenvalue decomposition of $(A - BK) = M\Lambda M^{-1}$,

$$\lim_{t\to\infty} x_t = \lim_{t\to\infty}(A - BK)^t x_0 = \lim_{t\to\infty} M\Lambda^t M^{-1}x_0\,,$$

The stability of a system means all components of the state vector x_t decaying to zero with time t, which is determined by the eigenvalues of $(A - BK)$. Define all eigenvalues of $(A - BK)$ as $\lambda_1, \lambda_2, ..., \lambda_n$:

If there is any eigenvalue with magnitude greater than one ($\exists i \in N, |\lambda_i| > 1$), the corresponding state component will grow exponentially with time and system is by definition unstable; If all eigenvalues have magnitude smaller than 1 ($\forall i \in N, |\lambda_i| < 1$), the system is stable; if there is an eigenvalue $|\lambda| = 1$ and all other eigenvalue no larger than 1, the system is defined to be marginally stable.

2.3 Linear Quadratic Gaussian (LQG)

The LQG problem is an extension of the LQR problem to a case where there is now Gaussian noise in the system, which means our system state evolves as:

$$x_{t+1} = Ax_t + Bu_t + w_t\,, \tag{27}$$

where $w_t \sim N(0, W)$ is a Gaussian noise

For this case the optimization equation in (6) is modified to:

$$\begin{aligned} \underset{u_0,\ldots,u_{N-1}}{\text{minimize}} \quad & J(u_0, ..., u_{N-1}, x_t) = \mathbb{E}[\sum_{\tau=t}^{N-1}(x_\tau^T Q x_\tau + u_\tau^T R u_\tau) + x_N^T Q_f x_N\,] \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t + w_t, \end{aligned} \tag{28}$$

Following the LQR derivation, let the optimal value function be at $i = t$ be $V^*(x_t) = x_t^T P_t x_t + \Sigma_t$. For $i = t-1$, $V^*(x_{t-1})$ can be obtained by backwards recursion as

$$\begin{aligned} V^*(x_{t-1}) &= \min_{u_{t-1}}(x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + \mathbb{E}[V^*(Ax_{t-1} + Bu_{t-1} + w_{t-1})]) \\ &= \min_{u_{t-1}}(x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + \mathbb{E}[(Ax_{t-1} + Bu_{t-1} + w_{t-1})^T P_t (Ax_{t-1} + Bu_{t-1} + w_{t-1}) + \Sigma_t]) \\ &= \min_{u_{t-1}}(x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + (Ax_{t-1} + Bu_{t-1})^T P_t (Ax_{t-1} + Bu_{t-1}) + \mathbb{E}[w_{t-1}^T P_t w_{t-1}] + \Sigma_t), \\ &= \min_{u_{t-1}}\left(x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + (Ax_{t-1} + Bu_{t-1})^T P_t (Ax_{t-1} + Bu_{t-1})\right) + Tr(WP_t) + \Sigma_t \end{aligned} \tag{29}$$

where $\mathbb{E}[w_{t-1}^T P_t w_{t-1}] = Tr(WP_t)$.

The derivative of (29) is the same as for LQR since the extra terms added are constants which are not a function of u_{t-1}, that is

$$\nabla_{u_{t-1}} V^*(x_{t-1}) = 2u_{t-1}^T R + 2(Ax_{t-1} + Bu_{t-1})^T P_t B = 0 \tag{30}$$

Hence the optimal control is,

$$u_{t-1}^* = -(R + B^T P_t B)^{-1} B^T P_t A x_{t-1} = -K_{t-1} x_{t-1} \tag{31}$$

Remarkably, the optimal control rule doesn't change even when we add noise to the system!

To get the optimal cost-to-go, we can substitute optimal control in (31) in equation (29):

$$\begin{aligned} V^*(x_{t-1}) &= x_{t-1}^T Q x_{t-1} + {u_{t-1}^*}^T R u_{t-1}^* + (Ax_{t-1} + Bu_{t-1}^*)^T P_t (Ax_{t-1} + Bu_{t-1}^*) + + Tr(WP_t) + \Sigma_t \\ &= x_{t-1}^T (Q + K_{t-1}^T R K_{t-1} + (A - BK_{t-1})^T P_t (A - BK_{t-1})) x_{t-1} + Tr(WP_t) + \Sigma_t \\ &= x_{t-1}^T P_{t-1} x_{t-1} + \Sigma_{t-1} \end{aligned} \tag{32}$$

where $P_{t-1} = Q + K_{t-1}^T R K_{t-1} + (A - BK_{t-1})^T P_t (A - BK_{t-1})$, $P_N = Q_f$, $\Sigma_{t-1} = Tr(WP_t) + \Sigma_t$ and $\Sigma_N = \vec{0}$

The infinite horizon derivation for the LQG follows accordingly from the LQR derivation.

3 Model Estimation

In computing the LQR and LQG in section 2, the model parameters are assumed to be known which makes computing the optimal controller straightforward. In cases where the system model is unknown, the LQR and LQG methods can still be used, the system model will first estimated before designing the optimal controller.

The most common method for model estimation is through Least Squares Estimation. There are different variants of this method but only two will be considered here: Ordinary Least Square (used off-line) and Recursive Least Square (used on-line)

3.1 Ordinary Least square

This method is used in estimating the system model using the input (control action) and output (state trajectory) data obtained from a linear dynamic system. Using the helicopter dynamics from section 1, assume the helicopter dynamics in (2) is unknown but we have obtained the input and output data by randomly driving the system resulting in $[x_t, u_t]_{t=0}^{T}$. Recall from (3), that the dynamics of a linear system can be written in the form:

$$x_{t+1} = Ax_t + Bu_t. \tag{33}$$

The least square estimation problem aims to find the unbiased estimator $\hat{A}$ and $\hat{B}$ of the unknown model A and B by solving the quadratic optimization problem:

$$(\hat{A}, \hat{B}) = \arg\min_{(A,B)} \sum_{t=0}^{T} \frac{1}{2} ||x_{t+1} - (Ax_t + Bu_t)||_2^2 , \tag{34}$$

To solve (34), let $z_t = \begin{bmatrix} x_t \\ u_t \end{bmatrix} \in \mathbb{R}^{n+p}$ and $\theta = [A,\ B] \in \mathbb{R}^{(n+p)\times n}$. Collecting the data for all time steps, we have:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix}, \quad Z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{T-1} \end{bmatrix}, \tag{35}$$

and (33) can be re-written as:

$$X = Z\theta. \tag{36}$$

Assuming $Z^T Z$ is invertible, the least square estimate of θ is given by [8]:

$$\hat{\theta} = (Z^T Z)^{-1} Z^T X. \tag{37}$$

3.2 Recursive least square

In the subsection above, the ordinary least squares is used in estimating the unknown model when the full data can be obtained beforehand, that is off-line. If the data comes in sequentially, for example, in an on-line application where the data is collected from a running system, we would like to recursively update the model as more data comes in. An ordinary least square estimate can be used repeatedly to obtain the estimate as

more data comes in but this is expensive due to the matrix inversion required. This gives rise to an algorithm called Recursive Least Square [4] which uses the matrix inversion lemma to reduce the computational burden.

From (35), suppose at time n we know $Z_n = z_0, \ldots, z_{n-1}$ and $X_n = x_1, \ldots, x_n$, then from (37):

$$\hat{\theta}_n = (\underbrace{Z_n^T Z_n}_{\Phi_n})^{-1} \underbrace{Z_n^T X_n}_{\delta_n} = \Phi_n^{-1}\delta_n \tag{38}$$

where $\Phi_n := Z_n^T Z_n = \sum_{i=0}^n z_i^T z_i$ and $\delta_n := Z_n^T X_n = \sum_{i=0}^n z_i^T x_i$.

When the next data arrives at time $n+1$, the updates will be as follows:

$$\Phi_{n+1}^{-1} = (\Phi_n + z_{n+1}^T z_{n+1})^{-1} = \Phi_n^{-1} - \frac{\Phi_n^{-1} z_{n+1}^T z_{n+1}^T \Phi_n^{-1}}{1 + z_{n+1}^T \Phi_n^{-1} z_{n+1}^T} \tag{39}$$

$$\delta_{n+1} = \delta_n + z_{n+1}^T x_{n+1} \tag{40}$$

$$\theta_{n+1} = \Phi_{n+1}^{-1}\delta_{n+1} \tag{41}$$

This process can be initialized with $\theta_0 = \mathbf{0}$ and $\Phi_0 = \alpha^{-1}\mathbf{I}$, where α is a very small positive number.

4 Recent Advances

In this section, we review some recent advances that have been made in the area of controlling linear systems with unknown system dynamics and providing guarantees on the optimality of the controller.

4.1 Linear Robust Control with Model Unknown

Paper [5] addresses the optimal control problem when the system dynamics are unknown. The proposed control procedure works as following:

a) estimate the unknown system dynamics model by least-square method,

b) quantify the least-square estimation error

c) design a robust controller using the error bound of system dynamics estimation.

The idea of first quantifying the system estimation error and then solving the optimal control problem by robust optimization is quite interesting and novel. Two major results are established in the paper: on the accuracy of least squares system estimation, and robust controller performance respectively. As we introduced in Section 3.1, least square method is widely used for unknown system estimation. Let's define the least square estimation of system matrix as $(\hat{A}, \hat{B})$. In order to quantify the least-square estimation error, we use only the final sample (x_T, x_{T-1}, u_{T-1}) of each trajectory (sample i.i.d. assumption).

Theorem 1. ***Least squares estimate accuracy***

Assuming the data is collected from a linear, time-invariant system initialized at $x_0 = 0$, using inputs $u_t \overset{i.i.d.}{\sim} N(0, \sigma_u^2 I_p)$ for $t = 1, ..., T$. Suppose that the process noise is $w_t \overset{i.i.d.}{\sim} N(0, \sigma_w^2 I_n)$ and we use only the final sample of each trajectory. With probability at least $1 - \delta$, the least squares estimator of system matrix satisfies:

$$||\hat{A} - A||_2 \leq \frac{16\sigma_w}{\sqrt{\lambda_{min}(\sigma_u^2 G_T G_T^* + \sigma_w^2 F_T F_T^*)}} \sqrt{\frac{(n + 2p\log(36/\delta))}{N}} \tag{42}$$

$$||\hat{B} - B||_2 \leq \frac{16\sigma_w}{\sigma_u}\sqrt{\frac{(n + 2p\log(36/\delta))}{N}} \tag{43}$$

where

$$G_T = [A^{T-1}B, A^{T-2}B, ..., B] \quad and \quad F_T = [A^{T-1}, A^{T-2}, ..., I]\,,$$

and,

$$N \geq 8(n+p) + 16\log(4/\delta)\,,$$

The above theoretical error bounds in (42) and (43) can only be computed if we know the true system matrices (A, B). However, this paper is trying to solve the optimal control problem when system dynamics are unknown. Therefore, the above theoretical bound may not be applicable and we need other methods to quantify the least square estimation error. The authors further proposed a bootstrapping method which provides an upper bound on the errors $\epsilon_A = ||A - \hat{A}||_2$ and $\epsilon_B = ||B - \hat{B}||_2$ from simulations.

With the least squares estimates $(\hat{A}, \hat{B})$ and error bounds (ϵ_A, ϵ_B) obtained from bootstrapping, a robust controller can then be designed. With high probability, we know the true system dynamics should be close to the LS estimation where $A = \hat{A} + \Delta A$, $B = \hat{B} + \Delta B$ and $||\Delta_A||_2 \leq \epsilon_A$, $||\Delta_B||_2 \leq \epsilon_B$. The objective of the robust LQR is to minimize the worst-case performance of the system given the (high-probability) norm bounds on the perturbations Δ_A and Δ_B

$$minimize \sup_{||\Delta_A||_2 \leq \epsilon_A, ||\Delta_B||_2 \leq \epsilon_B} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} E[x_t^* Q x_t + u_{t-1}^* R u_{t-1}] \tag{44a}$$

$$\text{s.t. } x_{t+1} = (\hat{A} + \Delta_A)x_t + (\hat{B} + \Delta_B)u_t + w_t \tag{44b}$$

Using System Level Synthesis (SLS) framework, a sub-optimal guarantee between the robust controller and the optimal LQR controller is given as follows:

Theorem 2. ***Sub-optimality guarantee*** *Let J_* denote the minimal LQR cost achievable by any controller for the dynamical system with system matrices (A, B) known, and let K_* denote the optimal controller. Let $(\hat{A}, \hat{B})$ be estimates of the system matrices such that $||\Delta_A||_2 \leq \epsilon_A$, $||\Delta_B||_2 \leq \epsilon_B$. Then, if K is the proposed robust controller. Then the relative error in the LQR cost is bounded as,*

$$\frac{J(A, B, K) - J_*}{J_*} \leq 5(\epsilon_A + \epsilon_B||K_*||_2)||\mathcal{R}_{A+BK_*}||_{H_\infty}\,, \tag{45}$$

with probability $1 - \delta$ provided N is sufficiently large.

This above result offers a guarantee on the performance of the proposed robust controller regardless of the estimation procedure used to estimate the system matrices. In this paper, they used least square estimation method. But in practice, this framework could be extended to work with different kinds of system estimation methods as long as the estimation error bound is provided.

4.2 Adaptive Control

A sub-field of control theory called *Adaptive Control* focuses on the design of controllers when the model parameters are unknown. In section 3, under the recursive least squares, no guarantees were given for the on-line estimated model and the subsequently designed linear quadratic (LQ) controller using the estimated model. Paper [6] provides these guarantees by deriving an error bound on the estimation error of the on-line model and a regret bound on the performance of the proposed on-line LQ controller.

4.2.1 Error Bound on Estimated Model

To estimate the model on-line, a high-probability confidence set is required for the unknown parameter matrix. Recall the modified linear system in where (36):

$$\Theta^T = [A,\ B] \qquad and \qquad z_t = \begin{bmatrix} x_t \\ u_t \end{bmatrix}, \tag{46}$$

and assume the uncertainties in the system are modeled as a Gaussian noise w_t such that we have a stochastic linear system, then the linear state-space equation can be written as:

$$x_{t+1} = \Theta^T z_t + w_{t+1} \tag{47}$$

To derive the confidence set, a self-normalization processes is used to estimate the least square estimation error:

$$e(\Theta) = \lambda \text{trace}(\Theta^T\Theta) + \sum_{s=0}^{t-1} \text{trace}((x_{s+1} - \Theta^T z_s)(x_{s+1} - \Theta^T z_s)^T) \tag{48}$$

Let $\hat{\Theta}_t$ be the ℓ^2-regularized least-squares estimate of Θ_*, with regularization parameter $\lambda > 0$:

$$\hat{\Theta}_t = \underset{\Theta}{\text{argmin}} e(\Theta) = (Z^T Z + \lambda I)^{-1} Z^T X, \tag{49}$$

where Z and X are the matrices whose rows are $z_0^T, \dots, z_{t-1}^T$ and $x_1^T, \dots, x_t^T$, respectively.

Theorem 3. *Let $(z_0, x_1), \dots, (z_t, x_{t+1})$, $z_i \in \mathbb{R}^{n+d}$, $x_i \in \mathbb{R}^n$, satisfy the linear model assumption in [6] some $l > 0$, $\Theta_* \in \mathbb{R}^{n\times(n+d)}$, $trace(\Theta_*^T\Theta_*) \le \mathcal{S}^2$ and let $\mathcal{F} = (\mathcal{F}_t)$ be the associated filtration. Consider the ℓ^2-regularized least-squares parameter estimate $\hat{\Theta}_t$ with regularization coefficient $\lambda > 0$. Let*

$$V_t = \lambda I + \sum_{i=0}^{t-1} z_i z_i^T$$

be the regularized design matrix underlying the covariates. Define

$$\beta_t(\delta) = \left(nL\sqrt{2log\left(\frac{det(V_t)^{1/2} det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{1/2}\mathcal{S} \right) \tag{50}$$

Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$,

$$trace\left((\hat{\Theta}_t - \Theta_*)^T V_t (\hat{\Theta}_t - \Theta_*)\right) \le \beta_t(\delta) \tag{51}$$

In particular, $\mathbb{P}(\Theta_*) \in \mathcal{C}_T(\delta), t = 1, 2, \dots) \ge 1 - \delta$, where

$$\mathcal{C}_T(\delta) = \left\{ \Theta \in \mathbb{R}^{n\times(n+d)} : \left\{ trace\left((\hat{\Theta}_t - \Theta_*)^T V_t (\hat{\Theta}_t - \Theta_*)\right) \le \beta_t(\delta) \right\} \le \beta_t(\delta) \right\} \tag{52}$$

This theorem shows that if the covariate matrix V_t is within a certain bound, then the deviation of the estimated model from the actual mode is bounded by $B_t(\delta)$.

4.2.2 Regret Bound on Performance Criterion

To define the regret of this system, consider the performance objective selected for the helicopter dynamics in section 1, let the cost at time t be:

$$c_t = x_t^T Q x_t + u_t R u_t. \tag{53}$$

The average expected cost over a time horizon T is given by:

$$J(u_0, u_1, \dots,) = \lim_{T\to\infty} \sup \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}[c_t] \tag{54}$$

Let J_* be the optimal (lowest) average cost which is the cost when an optimal controller that has the full information about the system dynamics is used. The *regret* up to time T of a controller which incurs a cost of c_t at time t is defined by

$$R(T) = \sum_{t=0}^{T} (c_t - J_*) \tag{55}$$

The algorithm for obtaining the on-line feedback controller is then as follows

Algorithm 1: Adaptive Algorithm for LQ Control

Input : $T,\ S > 0,\ \delta > 0,\ Q,\ L,\ \lambda > 0.$

Set $V_0 = \lambda I$ and $\hat{\Theta}_0 = 0$;
for $t := 0, 1, 2, \dots,$ **do**
  if $det(V_t) > 2\ det\ (V_0)$ **then**
    Calculate $\hat{\Theta}$ using equation (49);
    Find $\tilde{\Theta}$ such that $J(\tilde{\Theta}) \leq \inf_{\Theta\ in \mathcal{C}_t(\delta)\cap\mathcal{S}} J(\Theta) + \frac{1}{\sqrt{t}}$;
    Let $V_0 = V_t$;
  else
    $\tilde{\Theta}_t = \tilde{\Theta}_{t-1}$
  end
  Calculate u_t based on the current parameters, $u_t = K(\tilde{\Theta})x_t$;
  Execute control, observe new state x_{t+1} ;
  Save (z_t, x_{t+1}) into the dataset, where $z_t^T = (x_t^T, u_t^T)$;
  $V_{t+1} := V_t + z_t z_t^T$
end

Following this algorithm will lead to a regret bound given by the theorem below:

Theorem 4. *For any $0 < \delta < 1$, for any time T, with probability at least $1 - \delta$, the regret of Algorithm 1 is bounded as follows:*

$$R(T) = \tilde{O}(\sqrt{Tlog(1/\delta)}) \tag{56}$$

where the constant hidden is a problem depended constant

Remarks: It is unclear how computationally tractable step 5 in the algorithm 1 above will be since the computation is expected to be performed on-line and requires a search through the space for the model that satisfies the condition. Nevertheless, it is a step forward in this active research area and with time, we can expect a more developed result.

Furthermore, the regret bound derived in Theorem 4 is novel as it provides an explicit bound for an LQ problem. A recent paper [9] seems to be an extension of this result but for both a stationary and time-varying system though the regret bound derived is also $\tilde{O}(\sqrt{T}$ where $\tilde{O}(.)$ hides the constants and logarithmic factor. In this recent paper, a Thompson sampling-based learning algorithm is used in estimating the unknown system parameters as compared to using a high confidence probability bound. It will be interesting to do a comparison of these two papers to see which is more computationally tractable and under what assumption will the bounds hold.

References

[1] Astrm, Karl Johan, and Richard M. Murray. Feedback systems: an introduction for scientists and engineers. Princeton university press, 2010.

[2] Boyd, S. (2009). Lecture 3: Infinite horizon linear quadratic regulator. https://stanford.edu/class/ee363/lectures/dlqr-ss.pdf.

[3] Boyd, S. (2009). Lecture 10: Linear Quadratic Stochastic Control with Partial State Observation. Retrieved from https://stanford.edu/class/ee363/lectures/lqg.pdf.

[4] Leus, G., Veen, A. (2017). Lecture 9: Recursive Least Squares. Retrieved from http://ens.ewi.tudelft.nl/Education/courses/ee4c03/slides/10 rls.pdf.

[5] Dean, Sarah, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. "On the sample complexity of the linear quadratic regulator." arXiv preprint arXiv:1710.01688 (2017).

[6] Abbasi-Yadkori, Yasin, and Csaba Szepesvri. "Regret bounds for the adaptive control of linear quadratic systems." In Proceedings of the 24th Annual Conference on Learning Theory, pp. 1-26. 2011.

[7] How, J., Frazolli, E. (2010). Lecture 10: Feedback Control Systems. Retrieved from https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16 30F10 lec10.pdf.

[8] Boyd, S., Lall, S. (2015). Lecture 11: Least Squares. Retrieved from http://ee263.stanford.edu/lectures/ls.pdf.

[9] Ouyang, Yi, Mukul Gagrani, and Rahul Jain. "Learning-based Control of Unknown Linear Systems with Thompson Sampling." arXiv preprint arXiv:1709.04047 (2017).