Continuous Time LQR

Shobhit Saheb Dey
Autonomous Ground Vehicles, IIT Kharagpur

August 12, 2020

1 Linear System and the cost function

A linear system whose state space is n-dimensional(there are n parameters to fully(or sufficiently) describe the current state of system) is characterized by the equation.

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} \tag{1}$$

Here the $n \times n$ matrix $\mathbf{A}$ characterizes the system's dynamic properties. The solutions to this equation is:

$$\mathbf{X}(\mathbf{t}) = e^{\mathbf{A}t}\mathbf{X}(\mathbf{0}) \tag{2}$$

where $e^{\mathbf{A}t}$ is defined analogously by the Taylor expansion $e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2t^2}{2!} + \frac{\mathbf{A}^3t^3}{3!} \ldots$

So if we diagonalize $\mathbf{A}$ as $\mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ where $\mathbf{P}$ is the matrix with eigenvectors of $\mathbf{A}$ as columns and $\mathbf{D}$ is diagonal matrix of eigenvalues. Using this diagonalization the equation (2) changes to:

$$\mathbf{X}(\mathbf{t}) = \mathbf{P}\begin{pmatrix} e^{\lambda_1 t} & 0 & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & e^{\lambda_n t} \end{pmatrix}\mathbf{P}^{-1}\mathbf{X}(\mathbf{0}) \tag{3}$$

This shows that for any non-zero initial vector $\mathbf{X}(\mathbf{0})$, if any particular eigenvalue λ_i has positive real part then the corresponding eigenvector v_i will blow up to infinity with time.

Hence the primary aim of using a control system here is to somehow actuate the system to "virtually" change the system dynamics where all the eigenvalues will have negative real part.

If we mechanically design **m number of actuators** that have linear response to the system then the equation that characterizes it would be:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u} \tag{4}$$

Here obviously $\mathbf{u}$ is $m \times 1$ and $\mathbf{B}$ is $n \times m$ matrix. So if we explicitly set $\mathbf{u} = -\mathbf{K}\mathbf{X}$ then the equation (4) becomes

$$\dot{\mathbf{X}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{X} \tag{5}$$

Hence we have "virtually" changed the system's dynamics and if we properly choose the matrix $\mathbf{K}$, then we can "control" the system's eigenvalues to anywhere we want. This is the basic idea behind LQR.

Now the problem is how are we going to choose **K**. Being a $m \times n$ dimensional matrix, there are basically mn parameters to choose with an objective to control n eigenvalues, so manual tuning or any sort of gradient descent is out of question. Secondly, if we are transforming some highly unstable system to highly stable through some actuation, it would imply giving large amount of actuation which may be practically impossible or unfavourable due to various reasons like, fuel or power limits, design limits or uneasy driving experience(speaking in context of self-driving where large amount of acceleration or steer may be uneasy to passengers). Thus we need to put some cost over actuations too.
Suppose the final target state you want your system to reach is X_f then the quadratic cost function is:

$$J = \int_{t_0}^{\infty} \frac{1}{2}[(X - X_f)^T Q(X - X_f) + u^T Ru]dt \tag{6}$$

Here Q and R are diagonal matrices with all entries positive. Each diagonal entry determines how much weight will be given to the corresponding component.Notice J is a "functional" of $X(t)$.
So the control theory problem boils down to an optimization problem with a constraint as equation (4) with slight change that now it will be linearized about X_f instead of 0. So to solve this constraint optimization problem we define a costate vector $\lambda(t)$(analogous to lagrange multiplier) and define the lagrangian as:

$$L = \int_{t_0}^{\infty} (\frac{1}{2}((X - X_f)^T Q(X - X_f) + u^T Ru) + \lambda^T (A(X - X_f) + Bu - \dot{X}))dt \tag{7}$$

Instead of having a target state X_f, we can simplify the equation by "translating" the coordinate axes itself to X_f using the transformation $X \rightarrow X + X_f$ because obviously the dynamics and lagrangian should be invariant to where we keep our coordinate axes. This would make our life simpler to:

$$L = \int_{t_0}^{\infty} (\frac{1}{2}(X^T QX + u^T Ru) + \lambda^T (AX + Bu - \dot{X}))dt \tag{8}$$

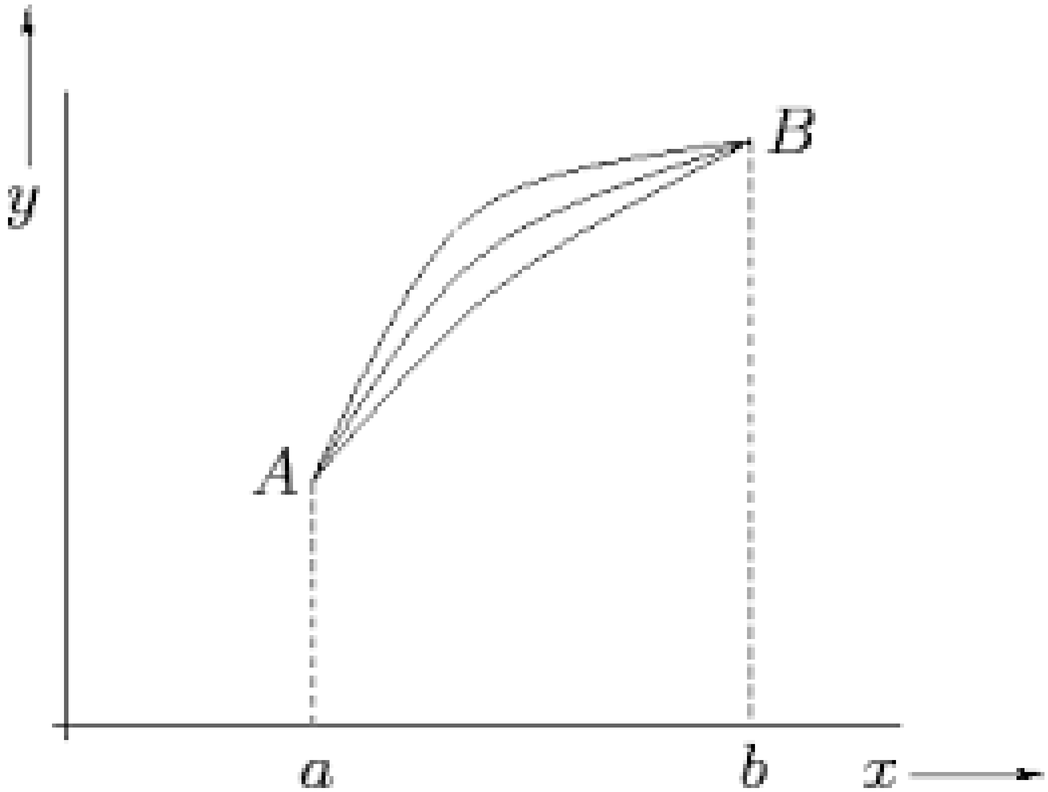

Figure 1: Variation of a function

2 Calculus of variations and some other stuff

Calculus of variations oversimplified: If if $f[y(x)]$(meaning that f is a "functional" of y which in turn is a "function" of x), in the domain (a,b) then $\frac{\delta f}{\delta y}\Big|_{y=y_0(x)}$ means the change in the value of f if the function y is slightly "varied" from $y_0(x)$ to $y_0(x)+\alpha.\epsilon(x)$. Slightly means that $\alpha \rightarrow 0$. Usually an additional constraint is imposed that $\epsilon(a)=\epsilon(b)=0$ so that variation at the end points of the domain is 0

As an example see Figure 1 where the mid curve (let it be our $y_0(x)$) is varied slightly. Now in this example if we consider a functional over this domain(could be anything for example: $f[y(x)] = \int\limits_a^b y^2 dx \quad or \quad \int\limits_a^b (sin(y) - y)dx$... just see how we require the values of y in the entire domain and not just at any specific point in domain (a,b) in order to evaluate f. This

is what makes it a "functional"). Now suppose that $f[y(x)]$ attains an extremum(maxima or minima) at $y \to y_0(x)$ then slight variations in $y(x)$ should not change the value of f, i.e $\frac{\delta f}{\delta y}\Big|_{y=y_0(x)} = 0$, completely analogous to what we find in conventional calculus.
Now coming to some standard derivatives in vector calculus:
In n-dimensional space, gradient w.r.t the vector $\mathbf{x}$ of a scalar $\mathbf{x}^T A\mathbf{x}$ is the vector $2A\mathbf{x}$
If we have a function in n-dimensions $f(\mathbf{x})$ then the differential change along the direction $\mathbf{dx}$ is $(\mathbf{dx})^T \boldsymbol{\nabla} f$ which is the dot product of $\mathbf{dx}$ and $\boldsymbol{\nabla} f$
So if we the differential change of $\mathbf{x}^T A\mathbf{x}$ in a direction $\mathbf{dx}$ is $2(\mathbf{dx})^T A\mathbf{x}$
Also $\boldsymbol{\nabla}\left(\mathbf{a}^T x\right) = \mathbf{a}$, so $\boldsymbol{\nabla}\left(\mathbf{a}^T Ax\right) = A^T\mathbf{a}$

3 Optimization

According to equation (8), the lagrangian is a functional of $X(t), \dot{X}(t)$ and $u(t)$. So we vary these 3 functions by arbitrary but infinitesimally small variations $\delta X\,(t), \delta\dot{X}\,(t)$ and $\delta u\,(t)$. In response to these variations, the differential change in the value of L using the relations in previous section are(remember that Q and R are symmetric):

$$\delta L = \int_{t_0}^{\infty} (\delta X^T\, QX + \delta u^T\, Ru + \delta X^T\, A^T\lambda + \delta u^T\, B^T\lambda - \delta\dot{X}^T\, \lambda)dt \tag{9}$$

To solve the last integral, we use integration by parts and the constraint from the previous section variation in the endpoint of the domain(which is (t_0,∞) here) is 0, i.e $\delta\dot{X}\,(0) = \delta\dot{X}\,(\infty) = 0$. Then we get

$$-\int_{t_0}^{\infty} \delta\dot{X}^T\, \lambda dt = \int_{t_0}^{\infty} \delta X^T\, \dot{\lambda} dt \tag{10}$$

Then equation (9) becomes after some rearrangement

$$\delta L = \int_{t_0}^{\infty} (\delta X^T\, (QX + \dot{\lambda} + A^T\lambda) + \delta u^T\, (Ru + B^T\lambda))dt \tag{11}$$

Since along the optimal trajectory in (X,u) space where L is minimum, δL should be zero in response to *any* arbitrary variations δX and δu. Hence it is possible if and only if:

$$QX + \dot{\lambda} + A^T \lambda = 0 \tag{12}$$

$$Ru + B^T \lambda = 0 \tag{13}$$

Now we play the trick of substituting $\lambda = Px$ where P is some $m \times n$ matrix(do not confuse it with the P matrix used in equation (3). Variable name P is used just to create some consistency with other literature you would read). The result of this trick using equation (13) is:

$$u = -R^{-1} B^T P X \tag{14}$$

Which is of the form $u = -KX$, exactly what we need and what we assumed our controller to be. Then just substituting $\lambda = Px$ in equation (12) and using equation(4) we get:

$$(Q + PA - PBR^{-1} B^T P + A^T P)X = 0 \tag{15}$$

Since this should be satisfied for any vector X, the matrix under braces should be 0.Then after some rearranging so that we get only P on LHS

$$P = (Q + PA - PBR^{-1} B^T P + A^T P)A^{-1} \tag{16}$$

This is called Continuous Algebraic Riccati equation or CARE. If we solve this equation for finding P, we automatically get a control policy using equation (14) and our control theory problem is solved. But the question is how do we solve this equation computationally. While we coded LQR we used the fixed point iteration method(because it resembles the problem of finding x such that $x = \phi(x)$ partly because we were still living in the nostalgia of 1st year mathematics course).