Online Optimal Control

Team communication document, The Controls Group, Institute for flight systems, DLR.
Author: Girish Chowdhary

Background:
An adaptive controller has been developed and demonstrated in HITL for the ARTIS helicopter [3]. This controller utilizes the online estimates of the system model using EKF to formulate new control gains at every time step. In this manner the varying parameters of the helicopter are accounted for. To update the gain matrix a recursive numerical algorithm for solving online the Riccati equation in optimal control was implemented in real-time. This document will look at the properties of convergence for this algorithm.

Abstract
The problem of designing an Extended Regulator is defined as designing an algorithm that can adjust the optimal regulator gains for a time-variant linear system appropriately to match the variations in system parameters with time. Due to conditions of positive definiteness on the unknown variable in the Riccati equation for quadratic optimal control it is possible to show that repetitive iterations of the Riccati equation will always converge to a steady state value. Furthermore if small changes in system parameters result in small changes in the extremal solution then recalculating the optimal solution using traditional backward integration or ordered Schur decomposition for every such change would be computationally inefficient. A more efficient approach could be to use the already available optimum solution as an initial guess to arrive at a close approximation of the neighboring optimal solution. This report aims to investigate in detail computationally efficient methods to arrive at close approximation of the neighboring optimal solution for the Riccati equation found in linear optimal control theory.

Nomenclature

\begin{align*}
A,B,C & \quad \text{State, input and observations matrices} \\
A_d,B_d & \quad \text{Updated state transition and input matrices} \\
K & \quad \text{Kalman filter gain matrix} \\
G & \quad \text{LQR gain matrix} \\
L,J & \quad \text{Scalar cost functions} \\
N & \quad \text{Time horizon for LQR recursion} \\
P & \quad \text{Covariance matrix of the state error} \\
Q,R & \quad \text{LQR state and input weighing matrices} \\
S & \quad \text{Riccati solution matrix} \\
FF^*,GG^* & \quad \text{Process and noise covariance matrices} \\
w(t),v(k) & \quad \text{Process and measurement noise functions} \\
f, g & \quad \text{Generic functions} \\
f_x & \quad \text{Partial derivative w.r.t variable x} \\
k & \quad \text{Discrete time index} \\
t & \quad \text{Time} \\
u & \quad \text{Control vector} \\
x & \quad \text{State vector} \\
y & \quad \text{State measurement vector} \\
z & \quad \text{Discretized State measurement vector} \\
\lambda & \quad \text{Lagrange Multiplier} \\
\Delta t & \quad \text{Discrete time step} \\
\text{Sub- and Superscripts} \\
H^T & \quad \text{Transpose} \\
R^{-1} & \quad \text{Inverse} \\
\hat{x} & \quad \text{Predicted variable} \\
\hat{x} & \quad \text{Corrected variable} \\
x_a & \quad \text{Augmented variable} \\
\text{Acronyms} \\
CARE & \quad \text{Continuous Algebraic Riccati Eq.} \\
DARE & \quad \text{Inverse} \\
MIMO & \quad \text{Predicted variable}
\end{align*}
Starting with a background to the problem, an overview of optimization theory leading to the derivation of the LQR TPBVP is provided. The conventional methods for solving the TPBVP are assessed before new ideas are proposed and validated with examples.

**Introduction**

The LQR problem addresses the problem of finding an optimal control input (Optimized against time to zero for individual states and the magnitude of control input) for a linear state space representation by minimizing a quadratic cost function. The benefit of using the LQR approach in control engineering is the guaranteed stability (as long as the system has no uncontrollable modes) and the possibility to optimize the control based on some specifications.

Recent applications in full authority adaptive control of MIMO time varying systems require that the LQR gains be calculated repetitively online in order to adapt to changing system parameters. Erdem in [9] analyses the problem of State-Dependent Riccati Equation (SDRE) Control, a methodology where the system matrices of a linear state-space representation are modeled as functions of the states of the plant. The system matrices are updated at each time-step depending on the current state of the system, control is achieved by solving the Riccati equation the corresponding time-step. The author has proposed the use of the Ordered Schur form based Hamiltonian Eigenvalue decomposition method [6] in order to solve the steady-state Riccati equation at every time step. Wan and Bogdanov demonstrate via simulation the control of a VTOL UAV using the SDRE method and nonlinear feedforward compensation in [9], they follow a similar method and solve the steady-state Riccati equation online. Both the works mention that solving the steady-state Riccati equation online is computationally extremely demanding. This suggests the need for computationally more efficient methods of addressing the problem of formulating optimal gain in adaptive control.

Bryson and Ho treat the problem of optimal control and estimation extensively in [1], they detail two methods for solving the Riccati equation arising in linear optimal control problem the first one being

In [3] we demonstrated the control of a VTOL UAV via online parameter identification and forward approximation of the solution to the Riccati equation. This report explains in detail our method to compute computational efficiency and have optimal control for time-varying systems. Instead of attempting to arrive at the steady-state solution of the Riccati equation by solving the DARE online using time consuming invariant subspace methods we exploit the fact that we have a close approximation for the optimal solution to the DARE for system parameters valid at the last time step available to us. We then implement an iterative method for arriving at a close approximation of the gain matrix that is valid for the current system parameters. Since our problem is fairly well-posed in mathematical terms we are able to extract extremely accurate approximations in as less as one iteration of the numerical method that we use to arrive at a solution to the DARE. Our method is computationally effective and has been demonstrated for the control of complex time variant systems in real-time.

In this report, starting with a background to the problem, an overview of optimization theory leading to the derivation of the LQR Two Point Boundary Value Problem (TPBVP) is provided. The conventional methods for solving the TPBVP are assessed before new ideas are proposed and validated with examples. Special consideration is given to numerical methods of addressing the Extended Regulator Problem

Problem Statement

Consider a discrete, linear, time varying state space representation of a dynamic system which is controllable and observable.

\[
x_{k+1} = A_k x_k + B_k u_k \quad k = 0, \ldots, N \quad x(t = t_0) = x(0)
\]

Optimal control is to be formulated by minimizing a discrete cost function of the form,

\[
j(u) = x_N^T Q_f x_N + \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right)
\]

Where Q and R denote the state and input weighing matrices and Q_f denotes the cost on final state deviation. The optimization results in a Two Point Boundary Value Problem (TPBVP). A well known and widely used solution to this problem is found using the *backward-sweep method* [1]:

I.e the solution to the following equation is found by starting from the final boundary condition at \( t = t_f \) and coming backwards in time to find the initial boundary condition at \( t = t_0 \)

\[
S_k = A_k^T \left[ S_{k+1} - S_{k+1} B_{k+1} (B_{k+1}^T S_{k+1} B_{k+1} + R)^{-1} B_{k+1}^T S_{k+1} A_{k+1} \right] A_{k+1} + Q \quad S_f = Q_f
\]

Then, using the appropriate value of S, the required LQR optimal feedback gain is calculated by the following relation,

\[
G_k = (B_{k+1}^T S_{k+1} B_{k+1} + R)^{-1} B_{k+1}^T S_{k+1} A_{k+1}
\]
S is said to have converged to a steady state solution when $S_{t+1} - S_t$ is zero. Note that Eq. 3 is to be solved backwards in time from $t_f$ to $t_0$ when in its presented form. For an offline case this would not pose any problems as complete knowledge of how a system changed in the past (i.e. change of system matrices w.r.t. time) is available beforehand. In an online application however, such an approach is not possible, since to calculate the gains at the current time step one must integrate backwards from a point in future. Since we are dealing with a time variant system operating in an uncertain environment it is not possible to account for all the unknown changes in system dynamics in advance. A solution to this problem could be to assume that the system holds its current value until a finite point in the future and then integrate backwards from that point in the future (starting from a predefined final value of $S$) to the current time point to obtain the steady state solution. However, this method is computationally expensive and ignores the information that has been accumulated as a result of previous iterations.

Alternatively it is possible to calculate the steady state LQR gain by solving the Discrete Algebraic Riccati Equation (DARE) using either invariant subspace methods such as Ordered Schur form based methods, Matrix Sign Function Method etc, or using iterative methods such as the first order gradient method, the Newton-Raphson method etc. Furthermore, due to the condition of positive definiteness on the Riccati variable $S$ in the optimal control problem it can be shown that repetitive iterations of the equation (3) will always converge to a steady state solution [7]. Consider now that an optimum solution for the system in equation (1) is known for epoch $k$. If at epoch $k+1$ the system undergoes small changes in parameters and if these small changes result only in small changes in the optimum solution the problem of finding the steady state solution is reduced to that of finding the perturbed solution for DARE.

Hence the problem of interest can be formulated as follows:

**Problem Statement:**

Given that the Riccati equation concerned with linear optimum quadratic control tends always to converge to a steady state value due to the constraint of positive definiteness of the Riccati variable $S$, investigate whether computationally more efficient methods can be realized for solving the Extended Regulator Problem. The Extended Regulator Problem is defined as the problem of designing a control architecture that adjusts optimal regulator gains to perturbations in the system parameters.

### 1. Basic optimization theory

Consider the optimization of a scalar cost function of $n+m$ parameters,

$$L(x_1, \ldots, x_n; u_1, \ldots, u_m)$$

Where the $n$ state parameters $(x_1, \ldots, x_n)$ are determined by the decision parameters $(u_1, \ldots, u_m)$.

Through a set of $n$ constraint relations,

$$f_i(x_1, \ldots, x_n; u_1, \ldots, u_m) = 0$$

$$\ldots$$

$$f_n(x_1, \ldots, x_n; u_1, \ldots, u_m) = 0$$

If we employ vector notation, and let $x$ denote the state vector, $u$ denote the control vector and $f$ denote the constraint vector then the problem could be stated in a more compact form as:

Find a decision vector $u$ that minimizes
Where the state vector $x$ is determined by control vector $u$ through a set of constraints:

$$f(x,u) = 0$$

(6)

A stationary point is one when the change in $L = 0,$ or $dL=0$ for arbitrary $du$ while holding $df = 0$:

$$dL = L_x dx + L_u du$$

(7)

And

$$df = f_x dx + f_u du$$

(8)

(Refer to Mathfacts for a clarification), note that $f_x$ and $f_u$ refer to partial derivatives of $f$ with respect to the $x$ and $u$ respectively.

We require that $df=0,$ and $f_x$ be non-singular since it determines $x$ from $u$ then,

$$dx = -f_x^{-1} f_u du$$

(9)

Hence,

$$dL = (L_u - L_x f_x^{-1} f_u) du$$

(10)

But since $dL=0,$ for any $du$ it follows,

$$L_u - L_x f_x^{-1} f_u = 0$$

(11)

These are $m$ equations, together with the $n$ equations of equation 6 they determine the $u$ and the $x$ vectors at the stationary point.

Another interesting approach would be to adjoin the constraints (equation 6) to the performance index (eq. 5) using $n$ undetermined multipliers (also known as Lagrange multipliers), $\lambda$. Then,

$$H(x,u,\lambda) = L(x,u) + \sum_{i=1}^{n} \lambda_i f_i(x,u)$$

$$= L(x,u) + \lambda^T f(x,u)$$

(12)

If $L=H,$ then $f(x,u)$ must be zero for some nominal value of $u.$ Then

$$dH = \frac{\partial H}{\partial x} dx + \frac{\partial H}{\partial u} du$$

(13)

Choosing the $\lambda$ vector such that it allows us to assess the change in $H$ w.r.t. change in $u$:

$$\frac{\partial H}{\partial x} + \lambda^T \frac{\partial f}{\partial x} = 0 \Rightarrow \lambda^T = - \frac{\partial L}{\partial x} \left( \frac{\partial f}{\partial x} \right)^{-1}$$

(14)

Since $x$ was found from equation 6,
\[ \partial L \equiv H = \frac{\partial H}{\partial u} du \]

And, \( \frac{\partial H}{\partial u} \) is the gradient of \( H \) w.r.t. \( u \) while holding \( f(x,u)=0 \). For a stationary point, \( dL \) must vanish for arbitrary \( du \), then

\[ \frac{\partial H}{\partial u} \equiv \frac{\partial L}{\partial u} + \lambda^T \frac{\partial f}{\partial u} = 0 \]

Hence the necessary conditions for the stationary value of \( L(x,u) \) are,

\[ f(x,u) = 0 \]
\[ \frac{\partial H}{\partial x} = 0 \]
\[ \frac{\partial H}{\partial u} = 0 \]
\[ H = L(x,u) + \lambda^T f(x,u) \]

These \( 2n+m \) equations define the quantities \( x, u \) and \( \lambda \) at the stationary value of \( L(x,u) \).

The sufficient condition for a local minimum is the validity of equation 15 and the positive definiteness of matrix of equation 16. A necessary condition is positive semidefiniteness of eq. 16.

\[ \frac{\partial^2 L}{\partial u^2} = H_{uu} - H_{ux}f_x^{-1}f_u - f_u^T \left( f_x^T \right)^{-1} H_{uu} + f_u^T \left( f_x^T \right)^{-1} H_{ux} f_x^{-1} f_u \]  \hspace{1cm} (16)

The above equations form the basis of optimization theory.

2. Optimization problem for dynamic systems

Consider a sequential formulation of the form:

\[ x(i+1) = f'[x(i),u(i)]; \quad x(0) \text{ given } \quad i = 0, ..., N-1 \]

The above system determines \( x(i) \) a vector with dimension \( n \) through the control vector \( u(i) \) a vector of \( m \) dimensions through \( n \) constraints \( f(i) \) which may be time variant. Now consider a performance index of the form:

\[ J = \Phi[x(N)] + \sum_{i=0}^{N-1} L[x(i),u(i)] \]

The performance index is formulated as a summation of the costs incurred at each time step \( I \) for \( i=0 \) to \( N-1 \) and \( I = N \). It is required to find a sequence of \( u(i) \) that minimizes (or maximizes) the performance index \( J \). We approach the problem by adjoining the performance index with a sequence of multipliers \( \lambda(i) \):
\[ J = \Phi[x(N)] + \sum_{i=0}^{N-1} [L'[x(i), u(i)] + \lambda^T(i + 1) \{ f'[x(i), u(i)] - x(i + 1) \}] \]  

(19)

Defining a scalar sequence \( H^i \)

\[ H^i = L'[x(i), u(i)] + \lambda^T(i + 1) \{ f'[x(i), u(i)] \} \quad i = 0, \ldots, N - 1 \]  

(20)

Equation 19 becomes after changing indices of the summation:

\[ \bar{J} = \Phi[x(N)] - \lambda^T(N) x(N) + \sum_{i=1}^{N-1} [H^i - \lambda^T(i) x(i)] + H^0 \]  

(21)

The differential changes in \( \bar{J} \) due to differential changes in \( u(i) \);

\[
d\bar{J} = \left( \frac{\partial \Phi}{\partial x(N)} - \lambda^T(N) \right) dx(N) + \sum_{i=1}^{N-1} \left\{ \frac{\partial H^i}{\partial x(i)} - \lambda^T(i) \right\} dx(i) + \left[ \frac{\partial H^i}{\partial u(i)} \right] du(i) \\
+ \frac{\partial H^0}{\partial x(0)} dx(0) + \frac{\partial H^0}{\partial u(0)} du(0)
\]  

(22)

To avoid calculating differential changes in \( dx(i) \) produced by a given \( du(i) \) we choose the multiplier sequence \( \lambda(i) \) such that:

\[
\begin{pmatrix}
\lambda^T(i) - \frac{\partial H^i}{\partial x(i)} \\
\end{pmatrix} = 0 \Rightarrow \lambda^T(i) = \frac{\partial L'}{\partial x(i)} + \lambda^T(i + 1) \frac{\partial f'}{\partial x(i)} \quad i = 0, \ldots, N
\]  

(23)

With boundary conditions,

\[ \lambda^T(N) = \frac{\partial \phi}{\partial x(N)} \]  

(24)

Which effectively reduces equation 22 to:

\[ d\bar{J} = \sum_{i=0}^{N-1} \frac{\partial H^i}{\partial u(i)} du(i) + \lambda^T(0) dx(0) \]  

(25)

Where \( \frac{\partial H^i}{\partial u(i)} \) is the gradient of \( J \) w.r.t \( u(i) \) while holding \( x(0) \) constant and satisfying eq. 17 whereas \( \lambda^T(0) = \frac{\partial H^0}{\partial x(0)} \) is the gradient of \( J \) w.r.t \( x(0) \) while holding \( u(i) \) constant and satisfying equation 17. If \( x(0) \) is known then \( dx(0)=0 \). [1]. Since for an extremum \( d\bar{J} \) must be zero,

\[ \frac{\partial H^i}{\partial u(i)} = 0; \quad i = 0, \ldots, N - 1 \]  

(26)

The above results can be summarized as follows,
Table 1: Summary of equations for finding a minimizing input for a sequential system

The problem presented in Table 1 is a Two Point Boundary Value problem (TPBVP) i.e. the boundary conditions must be satisfied at both ends of the time span. TPBVPs are considerably harder to solve compared to the relatively easier initial value problem. The reader is referred to Mathfact 2 for more information. The generic problem explained here with a fixed terminal time forms the basis of optimization theory for dynamic systems. The problem can be further extended to account for variables or functions of variables specified at a fixed terminal time, as well as variables or functions of variables specified at an unspecified terminal time (Including minimum time problem). Reference [1] treats these problems in considerable detail. Suffice is to mention that a solution to the optimum control problem for a generic time variant dynamic system can be obtained by solving the above mentioned TPBVP using iterative numerical methods ([2]) or by simplifying through assumptions as is demonstrated in the next section.

The equations of Table 1 are for sequential or discrete time formulation, they are easily extended to their continuous time counterparts by taking limit as the time increment between the steps goes to zero. Due to scope constraints we will not derive the equations for continuous systems in this section for the above problem, however this report contains derivation of the continuous domain equations for a closely related problem in the next section. Bryson [1] has exhaustive derivations for the interested reader.
3. Optimal Feedback control

In optimal feedback control we intend to find the optimizing control vector \( u^0(x,t) \) which is a function of time \( t \) and the current state \( x(t) \) with a unique optimal value of the performance index, \( J^0 = J^0(x,t) \). Where \( J^0 \) is dependent on the initial value and is known as the optimal return function [1]. The classical Hamilton-Jacobi theory allows us to find partial differential equations satisfied by the optimal return function \( J^0 \), Bellman extended this theory for multistage systems, the extended theory is known as “Dynamic Programming”. The Hamilton-Jacobi-Bellman (H-J-B) theory of dynamic programming is based extensively on the basic principles of the calculus of variations. The H-J-B theory can be briefly stated through the H-J-B equations as follows:

Consider an adjoined cost function of the form

\[
H = L[x,u,t] + \lambda^T \{ f[x,u,t] \}
\]

(27)

Then,

\[
\frac{\partial J^0}{\partial t} = H^0\left(x, \frac{\partial J^0}{\partial x}, t\right)
\]

where

\[
H^0\left(x, \frac{\partial J^0}{\partial x}, t\right) = \min u H\left(x, \frac{\partial J^0}{\partial x}, t\right)
\]

(28)

These equations indicate that \( u^0 \) is the value that minimizes globally the Hamiltonian \( H(x,\partial J^0/\partial x,u,t) \) holding \( x,\partial J^0/\partial x \), and \( t \) constant. The Euler Lagrange equations can be easily derived from the Hamiltonian Jacobi equations. The reader is referred to reference [1] for a detailed treatment:

\[
\dot{\lambda}^T = -\frac{\partial H}{\partial x} = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x}
\]

boundary conditions

(29A)

\[
\lambda^T(t_f) = \frac{\partial \phi}{\partial x(t_f)}
\]

And for an extremum

\[
\frac{\partial H}{\partial u} = 0 \quad t_0 \leq t \leq t_f
\]

(29B)

4. Optimization problem for Linear Systems

The H-J-B equation (eq. 28) does not yield for nonlinear systems of any practical significance ([1]), this prevents the development of exact explicit feedback guidance and control scheme for nonlinear systems. However an exact solution is available for linear systems with proper performance criteria. As is widely observed, linear systems adequately model most real world systems of engineering importance. The helicopter, although an essentially nonlinear system, can be adequately modelled by using number of linear systems linearized around specific trim points in the flight envelop. Hence optimal control, based around time variant linear systems optimized by minimizing a quadratic criterion is of prime importance in control applications.

Consider a time varying continuous linear system of the form,

\[
\dot{x} = A(t)x + B(t)u \quad x(t_0) \quad \text{given} \quad t_0 \leq t \leq t_f
\]

(30)

Girish Chowdhary
Where $x$ is the $n$ component state vector and $u$ is the $m$ component control vector. Consider a problem with a finite time horizon, $t_f$. We wish to bring the system from a known initial state $x(t_0)$ to a final state $x(t_f) \equiv 0$ at the terminal time $t_f$. While doing this we wish to define an acceptable margin for the control and an acceptable bound for the state deviation. We address this problem by minimizing a quadratic performance index, which has a quadratic form in the terminal state and a integral of quadratic forms in the state and the control:

$$J = \frac{1}{2} \left( x^T Q_f x \right)_{t=t_f} + \frac{1}{2} \int_{t_0}^{t_f} \left( x^T Q x + u^T Ru \right) dt$$  \hspace{1cm} (31)$$

Where $Q_f$ and $Q$ are positive definite matrices and $B$ is a positive semi definite matrix. These matrices are the tuning parameters in this problem a proper choice of these matrices will relate into meaningful satisfaction of the control constraints. Considering equations 29A and 29B:

$$H = \frac{1}{2} x^T Q x + u^T Ru + \lambda^T (Ax + Bu)$$  \hspace{1cm} (32)$$

Then

$$\dot{\lambda} = -Q x - A^T \lambda$$

and

$$0 = Ru + B^T \lambda \Rightarrow u = -R^{-1} B^T \lambda$$  \hspace{1cm} (33) \hspace{1cm} (34)$$

Substituting 33 in 30 and combining we have a linear two point boundary value problem:

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad x(t_0) \text{given} \quad \lambda(t_f) = Q_f x(t_f)$$  \hspace{1cm} (35)$$

The above problem depends on the initial condition $x(t_0)$, the coefficient matrix in the above equation is known as the Hamiltonian matrix.

4.1 Solution to the linear TPBV

We are concerned at this point with the time variant case; hence a steady state solution (or a stationary solution) may not exist. Our aim is to find a solution algorithm that is accurate, fast and computationally inexpensive, which gives us the required optimum control input at every time step. We start with considering the most straightforward way of finding a solution i.e. via the transition matrix. Note that in this section we treat the problem of optimization within a finite time horizon.

4.1.1 Solution by transition matrix:

To solve 35 we could use the method of linear superposition (mathfact 3) and determine $n$ linearly independent solutions to the $2n$ differential equation,

$$x^{(i)}(t) \quad \text{and} \quad \lambda^{(i)}(t), \quad i = 1,\ldots,n$$

Each solution satisfying the terminal boundary condition; hence:

$$\lambda^{(i)}(t_f) = Q_f x^{(i)}(t_f)$$

Determining unit solutions,
\[ x_j^{(i)}(t_f) = \begin{cases} 
1 & i = j; \\
0 & i \neq j; 
\end{cases} \quad \hat{\lambda}_j^{(i)}(t_f) = \left( Q_f \right)_{ji}. \]

Arranging these n solutions in columns gives us two transition matrices \( X(t) \) and \( \Lambda(t) \) with,

\[ X_{ji} = x_j^{(i)}(t), \quad \Lambda_{ji} = \hat{\lambda}_j^{(i)}(t) \tag{36} \]

and

\[ X(t_f) = I, \quad \Lambda(t_f) = Q_f \tag{37} \]

Then by superposition, if we knew \( x(t) \) we could write the solution at any time instant \( t \) as,

\[ x(t) = X(t)x(t_f) \quad \hat{\lambda}(t) = \Lambda(t)x(t_f) \tag{38} \]

At \( t_0 \) we can invert the above equations to obtain \( x(t_f) \) in terms of known \( x(t_0) \):

\[ x(t_f) = \left[ X(t_0) \right]^{-1} x(t_0) \tag{39} \]

Substituting Eq. 39 into Eq. 38:

\[ x(t) = X(t)[X(t_0)]^{-1} x(t_0) \quad \hat{\lambda}(t) = \Lambda(t)[X(t_0)]^{-1} x(t_0) \tag{40} \]

And substituting the value of \( \hat{\lambda}(t) \) in equation 34 we can find a relation for the optimum control input \( u(t) \) at time \( t \).

\[ u(t) = -G(t, t_0)x(t_0) \]

where

\[ G(t, t_0) = \left[ R(t) \right]^{-1} B^T(t) \Lambda(t) [X(t_0)]^{-1} \tag{41} \]

The above equations can be written in discrete sequential formulation, of sample time \( \Delta t \) such that \( t_{\text{new}} = t_{\text{old}} + \Delta t \) by denoting \( t_0 \) as the most recent sample time.

\[ x(k + 1) = X(k + 1)[X(k)]^{-1} x(k) \quad \hat{\lambda}(k + 1) = \Lambda(k + 1)[X(k)]^{-1} x(k) \tag{42} \]

With this formulation it is obvious that at every time step \( k \), to calculate the state at \( k+1 \), one must have the knowledge of the transition matrices \( X \) and \( \Lambda \) at \( k+1 \). The drawback of this approach in solving the TPBV is that the transition matrices could be ill conditioned for a large value of \( t_c \). Furthermore the eigenvalues of the Hamiltonian are symmetric around the imaginary axis (Butterfly pattern) hence there are always eigenvalues with positive real parts i.e. the adjoined system is unstable. An aggressive choice of \( Q \) and \( R \) matrices may even make the calculation of the transition matrix difficult for small values of \( t_c \). Vaddi et al. present in [4] a multi-stepping based method, which divides the time interval in smaller parts and hence allows stable computation of the transition matrix. They argue that this method is advantageous in terms of computational power as compared with the standard backward integration method described in the next section.

Girish Chowdhary
4.1.2 Solution by the sweep method, continuous case:

Taking limit of equation 41 as \( t-t_0 \) tends to zero, we have

\[
u(t) = -G(t)x(t)
\]

where

\[
G(t) = [R(t)]^{-1}B^T(t)\Lambda(t)[X(t_0)]^{-1}
\]

Now using equation 40 we can write the solution for \( \lambda(t) \) as:

\[
\lambda(t) = S(t)x(t)
\]

where

\[
S(t) = \Lambda(t)[x(t)]^{-1}
\]

and

\[
S(t_f) = Q_f
\]

Since we know the final value of \( S \), it is possible to sweep it backwards in time to find the initial value of \( S \). If in this way the correct initial value of \( S \) is found for a given optimization problem it would then be possible to integrate eq (35) forward as an initial value problem. Substituting the expression for \( S \) in eq. (35):

\[
\dot{S}x + S\dot{x} = -Qx - A^T Sx
\]

Substituting the value of \( \dot{x} \) in eq. (45) and using eq. (45),

\[
\left(\dot{S} + SA + A^T S - SBR^T B^T S + Q\right)x = 0
\]

Since \( x \neq 0 \),

\[
\dot{S} + SA + A^T S - SBR^T B^T S + Q = 0
\]

With the known terminal boundary condition,

\[
S(t_f) = Q_f
\]

If equation (47) is integrated backwards in time starting with terminal time \( t_f \) to the initial time \( t_0 \) it is possible to determine \( \dot{\lambda}(t_0) \) and is then possible to solve equation (35) forward in time. The control is then formulated by using the continuous feedback law [1]:

\[
u(t) = -G(t)x(t)
\]

where

\[
G(t) = [R(t)]^{-1}B^T(t)S(t)
\]

For each value of \( S(t) \). Clearly this approach requires that complete knowledge of the system matrices as functions of time be available along with the knowledge of initial state and assumption for the values of the tuning matrices, \( Q_f, Q \) and \( R \). Only with this knowledge can the backward integration take place resulting in a fully converged solution at \( t = t_0 \).
4.1.3 Solution by the sweep method, discrete case:

Similarly the backward sweep method can be proven for a multistage or discrete process. Assume a system of the form,

\[ x_{k+1} = A_k x_k + B_k u_k, \quad k = 0, \ldots, K \quad x_0 \quad \text{given}. \]  

(50)

Note that system and input matrices A and B are in their discretized form. With a cost function to minimize of the form,

\[
J = \frac{1}{2} x_N^T Q_N x_N + \sum_{i=0}^{N-1} \left[ \frac{1}{2} x_i^T Q_i x_i + \frac{1}{2} u_i^T R u_i \right]
\]

\[ i = 0, \ldots, N \]

The problem is to arrive at an optimizing u at each epoch by minimizing the cost between the time span \( t_0 - t_N \) and meeting the final cost constraint at \( i=N; N \leq K \), where the control input is defined as a linear function of the state:

\[ u_k = -G_k x_k. \]

The \( H_i \) sequence for this problem is,

\[
H_i = \frac{1}{2} x_i^T Q_i x_i + \frac{1}{2} u_i^T R u_i + \lambda_{i+1}^T \left[ A_i x_i + B_i u_i \right]
\]

Where the lambda sequence for this problem is found from equation 23,

\[
\lambda_i^T = \lambda_{i+1}^T A_i + x_i^T Q_i, \quad \text{and} \quad \lambda_N^T = x_N^T Q_N \quad i = 0, \ldots, N
\]

Substituting the following expression for lambda in the above equations a solution to this problem is obtained.

\[ \lambda_i = S_i x_i, \]

After some algebraic manipulations we arrive at,

\[
G_i = \left[ R_i + B_i^T S_{i+1} B_i \right]^{-1} B_i^T S_{i+1} A_i
\]

(51)

And

\[
S_i = A_i^T S_{i+1} A_i - A_i^T S_{i+1} B_i G_i + Q
\]

(52)

This backwards recursive method must be continued from \( i = N \) to \( i=0 \), using the value of S at \( i=0 \) we can then find the value of lambda at \( i=0 \). However we are only concerned with feedback control, which is realized through using the appropriate values of G at each epoch.

For real-time applications, however, changes in the system of time variant and/or state dependent systems operating in uncertain environments are not easily predicted beforehand, hence it is not feasible to use the backwards-sweep method for the calculation of optimal feedback gain.

An approach to this problem is to assume that the system holds its condition for a certain number of steps in the future and integrate backwards to find the gain. It is possible to employ a receding horizon...
approach to solve this problem in real time, however in the receding horizon approach only the gain term at \( i=0 \) is used and the rest are discarded. Computationally this may not be efficient if other solutions can be employed, moreover using gain term at \( i=0 \) for a stationary system relates directly to using the converged steady-state gain.

4.1.4 Steady-state solution, or stationary optimal control:
If the system parameters are time invariant it is possible that eq. (47) or eq. (52) will converge to a steady state value. The steady state Riccati equation can then be represented as algebraic Riccati equations by assuming \( \dot{S} = 0 \) for the continuous case and assuming \( S_{i+1} = S_i \) for the discrete case. The Continuous Algebraic Riccati Equation (CARE) is:

\[
SA + A^T S - SBR^T B^T S + Q = 0 ,
\]

and the Discrete Algebraic Riccati Equation (DARE) is:

\[
A^T SA - S - A^T SB(R + B^T SB)^{-1} B^T SA + Q = 0 .
\]

4.1.5 Solution for infinite operational time:
Consider now the problem of designing a feedback controller for a plant operating in real-time for indefinite intervals of time. We wish to keep the output of the plant within acceptable deviations from a reference output (regulator problem). Consider now that we are not interested in specifying strict constraints on the value of the S matrix at any given point of time, rather we are interested in arriving at the optimal solution for the regulator problem for the given state and input weighing matrices Q and R. Hence imposing a final cost \( Q_f \) is no longer required, furthermore for all real-time purposes we can set the length of the complete time interval to infinity. If we are dealing with a stationary regulator problem (the problem of designing a feedback controller for keeping a stationary or time invariant system within acceptable deviations from a reference state) it would be possible to set the final boundary condition \( S_f \) to zero and integrate eq. (47) backwards until \( S_0 \approx 0 \). In this case it can be shown that \( S(t) \rightarrow S(0) \), i.e. a steady state solution is arrived (Kalman 1968). However for time variant systems a definite steady state solution may not exist. Furthermore since changes in the system parameters cannot be predicted beforehand a backward integration method is not feasible. A solution to this problem can be found by dividing the time into finite time segments, assuming that the system parameters stay constant within those segments and then solving for the steady state solution of the CARE or DARE optimal for that time segment, this approach will be discussed in detail in section 5. However, greater insight into the functioning of the optimal control problem can be achieved if we analyze the problem for infinite operational time.

Consider now a cost function of the form,

\[
J = \frac{1}{2} \int_{t_0}^{t_f} (x^T Q_0 x + u^T R u) dt + \frac{1}{2} \int_{t_0}^{t_f} (x^T Q_f x) dt, \quad t = t_0, \ldots, t_f, \quad t_f \rightarrow \infty . (53)
\]

Where \( Q_0 \) and \( Q_f \) define S in the following manner:

\[
S(t_0) = Q_0, \quad S(t_f) = Q_f
\]

This new formulation allows specification of cost on the initial input separately. \( Q_0 \) could be found by solving the CARE for the appropriate values of A, B, Q and R matrices. Alternatively \( Q_0 \) could be used as a design variable in order to obtain desired stability near the initialization point. A clear condition here on \( Q_0 \) must be that it renders the system stable at time \( t_0 \), i.e.

Girish Chowdhary
\[ A_0 - B_0[R_0(t_0)]^{-1}B_0^T(t_0)S_0(t_0), \]

Must be a stable matrix that has no eigenvalues with nonnegative real parts. This puts a constraint of positive definiteness on the S matrix, and \( Q \geq 0, R \geq 0 \). Consider a generalized form of the cost function of \( \text{eq 51} \) in the form,

\[ J = \varphi[x(t_0),t_0] + \int_{t_0}^{t_f} L[x(t),u(t),t]dt + \varphi[x(t_f),t_f]. \quad (55) \]

Now, adjoining system equations 30 with to \( J \) with a multiplier function \( \lambda \) :

\[ \tilde{J} = \varphi[x(t_0),t_0] + \int_{t_0}^{t_f} \left[ H[x(t),u(t),t] + \dot{\lambda}^T(t)\{ f[x(t),u(t),t] - \ddot{x} \} \right]dt + \varphi[x(t_f),t_f]. \quad (56) \]

As in section 2, define a Hamiltonian \( H \) as follows:

\[ H[x(t),u(t),\dot{x}(t),t] = L[x(t),u(t),t] + \dot{\lambda}^T(t)\{ f[x(t),u(t),t] \}. \]

Integrating the last term of equation 54 by parts yields,

\[ \tilde{J} = \varphi[x(t_0),t_0] + \varphi[x(t_f),t_f] + \int_{t_0}^{t_f} \left[ H[x(t),u(t),t] + \dot{\lambda}^T(t)\{ f[x(t),u(t),t] - \ddot{x} \} \right]dt - \lambda^T(t_f)x(t_f) + \lambda^T_0(t_0)x(t_0). \quad (57) \]

Now taking limits as \( t_f \to \infty \):

\[ \tilde{J} = \varphi[x(t_0),t_0] + \lim_{t_f \to \infty} \left\{ \varphi[x(t_f),t_f] \right\} + \lim_{t_f \to \infty} \int_{t_0}^{t_f} \left[ H[x(t),u(t),t] + \dot{\lambda}^T(t)x(t) \right]dt \]

\[ - \lim_{t_f \to \infty} \left[ \lambda^T(t_f)x(t_f) \right] + \lambda^T_0(t_0)x(t_0), \]

And considering the variations in \( J \) due to variations in \( u(t) \) for fixed times \( t_0 \) and \( t_f \to \infty \):

\[ \delta J = \left[ \left( \frac{\partial \varphi}{\partial x} - \lambda^T \right) \delta x \right]_{t=t_0} + \lim_{t_f \to \infty} \left[ \left( \frac{\partial \varphi}{\partial x} - \lambda^T \right) \delta x \right]_{t=t_f} + \lim_{t_f \to \infty} \int_{t_0}^{t_f} \left[ \left( \frac{\partial H}{\partial x} + \dot{\lambda}^T \right) \delta x + \frac{\partial H}{\partial u} \delta u \right]dt. \quad (58) \]

Since it would be tedious to determine the variations in \( x \) produced by a given variation in \( u \) at time \( t \), we choose the Lagrange multiplier \( \lambda \) such that the coefficients of \( \delta x \) vanish in equation 56:

\[ \dot{\lambda}^T = -\frac{\partial H}{\partial x} = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x}, \quad (59) \]

With multiple boundary conditions.
\[
\lambda^T(t_0) = \frac{\partial \varphi}{\partial x(t_0)} \quad \text{and} \quad \lim_{t_f \to \infty} \left[ \lambda^T(t_f) = \frac{\partial \varphi}{\partial x(t_f)} \right].
\] (60)

Hence equation (58) then becomes,

\[
\delta J = \lim_{t_f \to \infty} \left[ \int_{t_0}^{t_f} \frac{\partial H}{\partial u} \delta u \, dt \right].
\] (61)

Clearly an extremum of the limit in equation (61) if it exists will occur only if \( \delta J \) is zero for an arbitrary \( \delta u(t) \); this is only possible when

\[
\frac{\partial H}{\partial u} = 0, \quad t_0 \leq t \leq t_f, \quad t_f \to \infty.
\]

Now, since we only approach but never reach infinity, the boundary condition at \( t=t_f \) is unnecessary and need not be specified explicitly in the design. Hence in summary we have the following differential equations that produce a stationary value of \( u \) at time \( t \) in order to minimize cost function of the form of equation (55),

\[
\dot{x} = f(x(t),u(t),t) \quad x(t_0) \quad \text{given} \quad t_0 \leq t \leq t_f, \quad t_f \to \infty
\]

\[
\dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda^T = -\frac{\partial L}{\partial x} - \lambda^T \frac{\partial f}{\partial x}
\]

Where \( u(t) \) is determined by,

\[
\frac{\partial H}{\partial u} = 0, \quad \text{or} \quad \left( \frac{\partial f}{\partial u} \right)^T \lambda + \left( \frac{\partial L}{\partial x} \right)^T = 0
\]

With boundary conditions

\[
x(t_0) \quad \text{given}
\]

\[
\lambda^T(t_0) = \frac{\partial \varphi}{\partial x(t_0)}
\]

and

\[
\left[ \lambda^T(t_f) = 0 \right]_{t_f \to \infty}.
\]

### Table 2 Proposed equations

Now for a linear system of the form of equation 30 with \( t_f \) tending to infinity, and the cost function of the form of equation (53), we have:

\[
\dot{\lambda}^T = -\frac{\partial H}{\partial x} ; \quad \lambda(t_0) = S_y x(t_0) \quad \text{and} \quad \frac{\partial H}{\partial u} = 0 \quad \text{where}
\]

\[
H = \frac{1}{2} x^T Q x + \frac{1}{2} u^T R u + \lambda^T (Ax + Bu).
\]

Performing the differentiations mentioned we obtain,
\[
\begin{bmatrix}
\dot{x} \\
\dot{\lambda}
\end{bmatrix} =
\begin{bmatrix}
A & -BR^{-1}BH^T \\
-Q & -A^T
\end{bmatrix}
\begin{bmatrix}
x \\
\lambda
\end{bmatrix}
\text{given } x(t_0) = x_0, \lambda(t_0) = Q_0 x(t_0)
\] (62)

It is important to note here that the above equation is a linear differential equation with homogeneous linear boundary conditions; hence the principle of superposition is applicable. Therefore, we can separate the solution of the above problem in two parts. The first being the solution from initial boundary to a finite time \( t \) in the future, and the other being solution from the final (infinite in this case) to time \( t \). Furthermore we choose the finite time \( t \) such that the system is stationary and the solution has converged to its natural steady state value. Since this natural steady state value would be the same for both solutions it is possible to simply add the solutions to form the complete solution to equation (62). Arguing on the same line we can conclude that integrating Equation (62) forward will approximate closely the backward solution. Furthermore, since all initial conditions are available equation (62) can be easily solved in the forward direction. Hence, it is then also possible to solve the following continuous time Riccati equation forward in time,

\[\begin{align*}
\dot{S} + SA + A^T S - SBR^T B^T S + Q &= 0, \\
S(t_0) &= Q_0.
\end{align*}\]

With the initial condition \( S(t_0) = Q_0 \). Since we have the initial value of \( S \) we can march the solution forwards in the following manner to obtain optimal control at each time step for a discrete linear representation of a dynamic system of the form,

\[x(k + 1) = A_x(k)x(k) + B_x(k)u(k) \quad k = 1, \ldots, \infty\]

In the following manner by propagating the solution matrix \( S \) to the next time step as follows

\[S(k + 1) = A^T(k)S(k)A(k) - A^T(k)S(k)B(k)(R + B^T(k)S(k)B(k))^{-1}B^T(k)S(k)A(k) + Q.\]

The optimal stabilizing gain at each time step can be then found as,

\[G(k) = -(R + B^T(k)S(k + 1)B)^{-1}B^T(k)S(k + 1)A(k).\] (63)

The logic behind this approach is clarified by considering the time transpose of the standard backward propagation algorithm. Hence rephrasing the LQR problem allows us to use it in a way that allows online implementation.

5. The Extended Regulator Problem

As mentioned previously the boundary conditions for eq. (35) are split at both ends of the finite time horizon problem. This is due to the choice of the cost function of the form of eq. (31) which allows specification of the final cost \( Q_f \). However eq. (47) converges rapidly to a steady state value due to the conditions on the \( Q, R \) and \( Q_f \) matrices. This indicates that as long as one is not extremely near the finite horizon \( t_f \) one can use the steady state solution which can be found by solving the algebraic Riccati equation:

\[\dot{S} = 0 = -SA - A^T S + SBR^T B^T S - Q \Rightarrow S(t) \rightarrow S_0 \text{ as } t_f - t_0 \rightarrow \infty,\] (64)

The Extended Regulator Problem is defined as the problem of designing a control architecture that adjusts optimal regulator gains to perturbations in the system parameters in real-time. An important property of the extended regulator problem is that no cost is prescribed for the final state, as it assumes infinite operational time. It is possible to tackle the extended regulator problem by solving the Discrete Algebraic Riccati Equation (DARE) at each epoch in order to obtain the steady-state optimal gain. The discrete formulation of the stationary or steady-state optimal linear quadratic control problem is,
Solution to this problem can be found numerically. Numerical methods for solving the DARE or the CARE are mainly classified in two categories:

1. Invariant subspace: E.g. the Hamiltonian eigen value decomposition method using the ordered Schur form [6] or the transition matrix method [4].
2. Iterative method: These methods employ iterative solutions to the two point boundary value problem. Some proven solution methods are:
   a. Shooting methods [2]:
      i. The first order gradient method [1],
      ii. The second order gradient method [1],
   b. Perturbation methods [1],
   c. Neighbouring extremal methods through backward propagation [1],
   d. Hermite natural iterative methods.

5.1.1 Solution to DARE using the Schur Method:

1. Define Hamiltonian matrix as
   \[
   \begin{bmatrix}
   A & -Q \\
   -B(\Delta t R)^{-1}B^T & -A^T
   \end{bmatrix}
   \]

2. Find Eigenvalues and eigenvectors of the Hamiltonian.
3. Partition the eigenvector matrix into four identical parts
   \[
   \begin{bmatrix}
   X_{11} & X_{12} \\
   X_{21} & X_{22}
   \end{bmatrix}
   \]
   where the eigenvectors corresponding to eigenvalues with positive real part are in the left partition.
4. S can now be found as
   \[
   S = -X_{11} \cdot X_{21}^{-1}
   \]

This method converges to the correct solution without reliance on iteration, and is known to be numerically stable if a numerically stable eigenvalue finding algorithm is used. The drawback of this method lies in the selection and implementation of the eigenvalue and eigenvector finding algorithm and the effort in sorting the eigenvectors. In general the numeric effort can be quite demanding for real-time applications.

5.1.2 Hermite Natural Iterative Method

This method ensures the stability of the numeric algorithm by posing certain constraints on the initial guess to the solution. The most important constraint is that the starting solution S, must be positive definite. The iterations are then described as,

\[
\begin{align*}
S_i &= Q + A^T(BR^{-1}B^T)^{-1}A \\
S_{i+1} &= Q + A^T((S)^{-1} + BR^{-1}B^T)^{-1}A \quad i \geq 1
\end{align*}
\]

The convergence of the above method is proved via induction in [7]. The benefit of using iterative methods in real-time online optimal control is amplified if the following condition is met by the discrete time-variant plant:
- Condition: Small changes in system parameters result only in small perturbations to the optimum solution. I.e. the problem is well-conditioned.

If this condition is met then optimal solution before plant parameter perturbation would be an extremely close initial guess for the numerical algorithm. The resulting reduction in number of iterations alleviates computational load on the processor. An interesting outcome of this approach is that it can now be argued that the forward integration method proposed in 4.1.5 is nothing but a one iteration implementation of the Hermite Naturally Converging algorithm. As results in Section 7 demonstrate, this implementation indeed approximates the steady solution very closely and efficiently.

6. Duality Considerations
It has been observed by various noted authors that there exists a definite duality between the filtering problem and the control problem [1],[5]. The proposed duality in these two texts can be extended to account for the more generic extended regulator problem as opposed to the comparison between a finite horizon control problem and a noisy filtering problem as stated in the texts. Consider the least squares filtering problem for a state space representation of a dynamic system of the form of equation 30 with a measurement equation of the form:

\[ z = Cx + \nu \]
\[ E(\nu) = 0 \]
\[ E(\nu^T) = R \]

Where R is the measurement covariance matrix of the white Gaussian noise vector \( \nu \).

The solution to the estimation problem is given as follows [1],[3],[5]:

\[ \dot{x} = Ax + B\bar{w} + PC^Tz^T R^{-1}(z - C\dot{x}), \quad \dot{x}(0) = 0 \]
\[ \dot{P} = AP + PA^T + BQB^T - PC^T R^{-1} CP \quad P(0) = P_0 \]

Where Q and R denote the noise covariance matrices of the state and measurement noises respectively, in case of a noise free estimation problem these matrices could be viewed as state and measurement weighing matrices.

The duality between the control and the estimation problem is evident in one compares equation 63 with equation 47. It is clearly seen that both equations are in the form of a Riccati equation are duals of each other if one replaces S with P, and associates the appropriate value of Q and R matrices and reverses time. A table discussing the duality of the noise free control problem and the filtering is presented below.

<table>
<thead>
<tr>
<th>Control Problem</th>
<th>Estimation Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B^T )</td>
<td>( C )</td>
</tr>
<tr>
<td>( C^T )</td>
<td>( G )</td>
</tr>
<tr>
<td>( Q )</td>
<td>( Q )</td>
</tr>
<tr>
<td>( R )</td>
<td>( R )</td>
</tr>
<tr>
<td>( S(t_f) )</td>
<td>( P(t_0) )</td>
</tr>
<tr>
<td>( A )</td>
<td>( A^T )</td>
</tr>
<tr>
<td>( t_f-t_0 )</td>
<td>( t_0-t_f )</td>
</tr>
</tbody>
</table>

Table 3, Duality between control and estimation problem,

Now considering the results of section 4.1.5, it can be argued that since the reversal of time is no longer required and complete duality is established between the optimal noisy estimation problem and the noise free regulator problem. Furthermore the similarities between the Riccati solution matrix in
optimal control problems and the covariance matrix in optimal estimation problems are indicative of a closer duality. Further explorations in this area are required to exploit this fact in the area of non-linear control.

7. Examples

In this section the convergence of the forward integration algorithm is demonstrated. We start with a simplified, unstable, continuous time variant, linear system of the form,

\[ \dot{x} = A(t)x + B(t)u \]

We consider a time frame of 1000 steps and transform the above system to its discrete form with a time step of 0.01 seconds. We simulate time dependency by continuously varying the system in a linear way for some predefined time period. The time history of the system matrix A and B is presented in Figure 1 Time histories of system matrices. Whereas Figure 2 Comparison of gains calculated using forward and backward methods compares the gains calculated using the backward sweep method, the steady state gains at each time step and the forward integration method.

![Figure 1 Time histories of system matrices](image1)

![Figure 2 Comparison of gains](image2)
Figure 2 Comparison of gains calculated using forward and backward methods

Clearly from Figure 2 it is seen that there is no difference between the values of the gains of the forward and backward integration method as long as one is not near the initial boundary or the finite horizon. In Figure 3 Closeup on the behaviour of Gain and Riccati solution presents the behavior of the optimal gain and the Riccati solution variable S in the initial few steps as well as the final few steps. , for this example the initial constraint on S i.e. $Q_0$ was set at 10, a value that renders the system stable at initial time $t=0$, alternatively one could just use the steady state Riccati solution value for initialization. The final constraint on S, i.e. $Q_f$ was also set to 10.

Figure 3 Closeup on the behaviour of Gain and Riccati solution
Behavior of the forward integration solution: The forward integration solution enforces a cost function of the form of equation 51, hence an initial constraint on the Riccati solution variable $S$ is enforced. It is seen that the Riccati solution as well as the optimal gain quickly converge to a steady value, changing only when the system itself goes under physical changes. In the final few steps it is seen that the forward integration method simply behaves as if the finite boundary is not reached, and hence the final constraint on the Riccati solution variable $S$ is not enforced.

Behavior of the backward sweep solution: The solution obviously starts from $t = t_f$, which is a finite value, $t_f = 1000$ for this problem. The final constraint on $S$ ($Q_f = 10$) is met by the backward sweep method and hence the final gain value is not the same as the steady state gain value. As no constraint on the initial value of $S$ was forced for the backward sweep method, the backward sweep method holds the steady state value at $t = 0$.

Behavior of the steady state solution: The steady state solution uses the solution to DARE found by minimizing a cost function of the form at each time step:

$$J = \int_0^\infty (x' Q x + u' R u) \, dt.$$  

The steady state gain does not enforce any initial or final constraints on the Riccati solution variable $S$ and hence the optimal gain $G$. The steady state solution was solved via the Schur ordered form for this example. Equivalently one may use any appropriate numerical algorithm as depicted in section 5.1.2.

Hence we can conclude that as long as we are not interested in a finite horizon based optimization forward integration method which minimizes a cost function enforcing a specific cost constraint on the initial state deflection to obtain the initial gain matrix would result in an optimal solution. The method presented in section 4.1.5 has been implemented for the ARTIS adaptive control system and have been proven in simulation in [3] [“Control of a VTOL UAV via online parameter identification”, Chowdhary G, Lorenz S, AIAA GNC 2005, San Francisco, USA.]

8. Conclusion

The Extended Regulator Problem was posed as the problem of designing a control architecture that adjusts optimal regulator gains to perturbations in the system parameters in real-time. Two methods of addressing the Extended Regulator Problem while maintaining numerical efficiency have been presented in this report:

1. By approximating the solution to the optimal control problem via forward integration of equation (52).
2. By solving the DARE through iterative numeric methods at every time step using the optimal solution from the previous step as the initial guess for the numeric method.

Results in section 7 clearly suggested that the forward integration method approximates extremely well the steady-state solution. Furthermore it can be clearly seen that the forward integration method is nothing but a single iteration implementation of Hermite iterative numeric solution method described in section 5.1.2. It can then be concluded that since convergence of the Riccati equation in the optimal control problem is always guaranteed due to the condition of positive definiteness of the Riccati variable $S$, forward iterations of the Riccati equation will result in converged solution. Furthermore, it is possible to use the optimal solution from the last time step as an initial guess in iterative numeric methods. This approach reduces convergence time dramatically for systems in which small changes in initial conditions result only in small perturbations to the optimal solution (or the problem is well-conditioned). The methods presented in this report have been demonstrated for control of a VTOL UAV in [3]. In future the author would like to extend the scope of this work by investigating more
closely the duality between the optimal control and the filtering problem in order to arrive at a common methodology for addressing the problem of Extended Gaussian Regulators.

References:


Mathfacts:

Mathfact 1. The Total Derivative
The total derivative of a function of multiple variables can be thought of as a derivative that takes all indirect dependencies into account or the sum of all partial derivatives with respect to all the variables in the function. In the latter case it is treated as an differential operator of the form:

\[ df(p_1,...,p_n) = \sum f_i dp_i \]

Where \( f_i \) denotes the partial derivative of \( f \) w.r.t. \( p_i \). One could interpret the total derivative as a relation that gives the total error (or change) in \( f \) due to error (change) in parameters \( p \). Also interesting is to note that the total derivative forms a linear approximation of the function at the point that it is taken.

Mathfact 2. The Two Point Boundary Value Problem

Consider an ODE of the form,

\[ y'_i = g_i(x, y_1, y_2, ..., y_n); \quad i = 1...N \]

For which the boundary conditions must be satisfied at two distinct points, say at point 1 we must satisfy \( n1 \) boundary conditions:

\[ B_{1j}(x_1, y_1, y_2, ..., y_n); \quad j = 1, ..., n1 \]
And n2=N-n1 boundary conditions at point 2:

\[ B_{2k}(x_2, y_1, y_2, ..., y_n); \quad k = 1, ..., n2 \]

Clearly the solution to the above problem must satisfy both the boundary conditions. This indicates that marching the solution forward by starting from a boundary condition may not guarantee that the other boundary condition is also met! Two methods of solving this problem exist. The first being the Shooting method, in which we begin with a guess of the initial values, integrate the ODE and then adjust the guess of the initial values so as to meet the boundary conditions at the end. This method is by far the most common method. The second method is called as the relaxation method, in which the differential equations are converted to finite difference equations the solutions space is discretized and meshed. The iterations then attempt to “relax” the solutions at each mesh point in order to simultaneously satisfy the finite difference equations and the boundary conditions. Detailed information on the solution to the TPBVP can be found in [2].

**Mathfact 3. Solution to the linear ordinary differential equation**

Consider an equation of the form,

\[ \dot{x} = A(t)x + B(t)u \quad x(t_0) \quad \text{given} \]

Where A and B are n*n and n*m time varying matrices. A solution to the above equation can be found by using linear superposition, i.e. by knowing that a solution to the linear ODE is a linear combination of its unit solutions. Assume that we apply a unit input to the \( j^{th} \) integrator at time \( \tau \), then let \( \phi_j(t_1, \tau) \) designate the response of the \( i^{th} \) integrator due to the unit impulse on input of the \( j^{th} \) integrator at time \( t_1 \). Doing this for all \( i \) and \( j \) will result in a n*n matrix \( \phi(t_1, \tau) \), and doing the whole process for time \( t_0 \) to \( t_1 \) will result in the complete time history of matrix \( \phi(t_1, \tau) \). Then for the arbitrary time interval \( t_1 > t_0 \) using linear superimposition we have [1],

\[ x(t_1) = \phi(t_1, t_0)x(t_0) + \int_{t_0}^{t_1} \phi(t_1, \tau)G(\tau)u(\tau)d\tau \phi_0(t_1, \tau) \]