MARIST
UNIVERSITY

Quantum Computing in Time-varying Linear Quadratic Regulator(LQR) problem.

Malick Ndiaye, Casimer DeCusatis, Augusto Gonzalez

November 3, 2025

What is Quantum computing?

Quantum computing is a type of computation that uses the principles of quantum mechanics such as superposition and entanglement.

Superposition-Entanglement

- Classical computers use classical bits: 0 or 1. However, quantum computers use qubits or quantum bits which can be in the quantum state of 0, 1 or a superposition of both at the same time.
- Entanglement in quantum computing is a phenomenon that allows the quantum states of two or more qubits to become linked, making impossible to describe one independently of the others.

Consequence

The consequence of these two principles (superposition and entanglement) is the potential for quantum computers to solve very complex problems in a short amount of time that are intractable for even the most powerful classical computers.

Mathematical algorithms that utilize quantum computing?

- Shor's Algorithm for integer factorization.
- Glover's Algorithm for unstructured search problems
- Deutsh-Jozsa Algorithm for determining whether an unknown function is constant or balanced.
- The HHL Algorithm.

The HHL Algorithm

This algorithm was developed by Aram Harrow, Avinatan Hassidin and seth Lloyd.
It deals with linear system $Ax = b$ where A is a $N \times N$ matrix.
This quantum algorithm approximates the solution in approximately $O(logN)$ time where A is a sparse matrix and Hermitian.
The best classical algorithms we have take approximately O(N) time.

Hybrid HHL

The Hybrid HHL is a quantum algorithm that modifies the original HHL algorithm by incorporating classical computing steps to make it more efficient and applicable on current quantum hardware.

The Enhanced Hybrid HHL

This algorithm uses a higher precision quantum estimates of the eigenvalues relevant to the linear system and a new classical step to guide the eigenvalue inversion part of the hybrid HHL.

Optimal Control Statement

A basic optimal control problem $(\mathcal{P})$ can be stated as follows: Given the system of differential equations along with an initial condition,

$$(S)\ \frac{dx}{dt} = f(x(t), u(t)), \quad x(t_0) = x_0$$

where $x(t)$ is the state of the system, $x(t) \in \mathbb{R}^n$, and $u(t) \in \mathbb{R}^m$ is the control.
The goal is to find a control $u(t)$ over $[t_0, t_f]$ which for any x_0, minimizes the cost function

$$J(x, u) = h(x_{(t_f)}) + \int_{t_0}^{t_f} L(x(t), u(t))dt$$

$L(x(t), u(t))$ is called the running cost and $h(x(t_f))$ is called the terminal cost

Necessary condition. The Pontryagin's maximum Principle

If $u^*(t)$ and $x^*(t)$ are optimal for the problem (P), then there exist a function $\lambda(t) = \begin{pmatrix} \lambda_1(t) \\ \lambda_2(t) \\ . \\ . \\ . \\ \lambda_n(t) \end{pmatrix}$ and a function H defined as:
$H(x(t), u(t), \lambda) = \lambda^T f(x, u) - L(x, u)$ that satisfy the three properties.
$a) H(x^*(t), u^*(t), \lambda(t)) \geq H(x^*(t), u(t), \lambda(t))$
for all control u at each time t.
$b) \frac{d\lambda}{dt} = -\nabla_x H(x^*, u^*, \lambda)$(Adjoint equation)
c)$\lambda(t_f) = \nabla_x h(x(t_f))$(Transversality condition)
λ^T represents the transpose of λ and H is called the Hamiltonian.

Sufficient Condition

Let $U(x_0)$ be the set of admissible controls of u and X an open subset of $\mathbb{R}^n$. If there exists a function $J_1 : X \to \mathbb{R}$ of class C^1 such that the three statements below are true
i)If $u \in U$ generates a solution $x(t)$ and $x(t) \in X$ for all $t \in [t_0, t_1]$, then $lim_{t \to t_1} J_1(x(t)) \leq lim_{t \to t_1^*} J_1(x^*(t)) = 0$, for some $t_1^* \geq t_1$
$ii)L(x^*(t), u^*(t)) + grad^T J_1(x^*(t)) f(x^*(t), u^*(t)) = 0$ for all $t \in [t_0, t_1^*)$ for some $t_1^* \geq t_1$
$iii)L(x, u) + grad^T J_1(x) f(x, u) \geq 0$ for all $x \in X$ and $u \in U$.
Then the control $u^*(t)$ generating the solution $x^*(t)$ for all $t \in [t_0, t_1^*]$ with $x^*(t_0) = x_0$, is optimal with respect to X.

The time-varying Linear-Quadratic Regulator(LQR)

We suppose that

$$f(x(t), u(t)) = A_1(t)x(t) + B_1(t)u(t)$$

$$L(x(t), u(t)) = \frac{1}{2}(x^T(t)Q(t)x(t) + u^T(t)R(t)u(t))$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A_1 \in \mathbb{R}^{n \times n}$, $B_1 \in \mathbb{R}^{n \times m}$, $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$.
The matrix R is symmetric that is $R^T = R$ and positive definite that is $x^T Rx > 0$ if $x \neq 0$.
The matrix Q is symmetric and positive semi-definite that is $x^T Qx \geq 0$.
The functions $A_1(t)$, $B_1(t)$, $Q(t)$, and $R(t)$ are of class C^1.

Applications

- Aircraft Control
- Robotics
- Power Systems
- Chemical Plants
- Manufacturing
- Automotive Control

Solution to the LQR-The Riccati Equation

We can use the Pontryagin's maximum principle to find $u(t)$.
If the terminal cost $h(x(t_f)) = 0$, then the control $u(t)$ is given by

$$u(t) = -R^{-1}(t)B_1^T(t)P(t)x(t)$$

where $P(t)$ is a solution of the Riccati equation

$$\frac{dP(t)}{dt} = P(t)B_1(t)R^{-1}(t)B_1^T(t)P(t) - A_1^T(t)P(t) - P(t)A_1(t) - Q(t)$$

satisfying the initial condition $P(t_f) = 0$.

General Form of the Matrix Riccati Equation

The Matrix Riccati equation can be written as:

$$\frac{dY(t)}{dt} = Y(t)A(t)Y(t) + Y(t)B(t) + C(t)Y(t) + D(t)$$

where $Y \in \mathbb{R}^{n\times m}$, $A(t) \in \mathbb{R}^{m\times n}$, $B(t) \in \mathbb{R}^{m\times m}$, $C(t) \in \mathbb{R}^{n\times n}$ and $D(t) \in \mathbb{R}^{n\times m}$.

Theorem

If $B(t) = 0$, $m = n$ and $A(t)$ is invertible, then the matrix Riccati equation can be turned into the second order matrix linear differential equation.

$$V^{''}(t) - (A(t)C(t)A(t)^{-1} + A(t)^{'}A(t)^{-1})V^{'}(t) + A(t)D(t)V(t) = 0$$

using the change of variable

$$Y(t) = -A(t)^{-1}V(t)^{'}V(t)^{-1}$$

where $V(t)$ is invertible.

Riccati equation in LQR

$$\frac{dP(t)}{dt} = PB_1R^{-1}B_1^TP - A_1^TP - PA_1 - Q(t)$$

$$P(t_f) = 0.$$

We assume that $R(t) = A_1(t) = I$ where I represents the $n \times n$ identity, Choosing $B_1(t)$ and $Q(t)$ such that $B_1(t)B_1^T(t)Q(t) = I$ and $2I - (B_1(t)B_1^T(t))'(B_1(t)B_1^T(t))^{-1} = -S$ where S is a constant diagonal matrix.

$$S = \begin{bmatrix} \alpha_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \alpha_n \end{bmatrix}$$

The change of variable

The change of variable $P(t) = -(B_1(t)B_1^T(t))^{-1}V'(t)V^{-1}(t)$ yields to the second order Matrix linear differential equation

$$V''(t) - SV'(t) - V(t) = 0 \quad V'(t_f) = 0$$

which in turn leads to the linear system:

$$H(t)\begin{bmatrix} X_{11}(t) \\ \vdots \\ X_{nn}(t) \end{bmatrix} = \begin{bmatrix} X_{11}(t_f) \\ \vdots \\ X_{nn}(t_f) \end{bmatrix}$$

where $V = (V_{ij}), 1 \leq i, j \leq n$

$$X_{ii}(t) = \begin{bmatrix} V_{ii}(t) \\ V'_{ii}(t) \end{bmatrix}$$

$$H(t) = \begin{bmatrix} e^{-M_1(t-t_f)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{-M_i(t-t_f)} \end{bmatrix} \text{ and } M_i = \begin{bmatrix} 0 & 1 \\ 1 & \alpha_i \end{bmatrix}$$

Experimental Result

To test the algorithm's performance, we used the following example.
$\frac{dx(t)}{dt} = A_1(t)x(t) + B_1(t)u(t)\; t \in [0, 1]$ so $t_f = 1$

$$J(x, u) = \frac{1}{2}\int_0^1 (x(t)^T Q(t)x(t) + u(t)^T(t)R(t)u(t))dt$$

where

$$A_1(t) = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_1(t) = \begin{pmatrix} e^{t/2} & -e^{t/2} \\ e^{t/2} & e^{t/2} \end{pmatrix}, \quad R(t) = I,$$

$$Q(t) = \begin{pmatrix} \frac{1}{2}e^{-t} & 0 \\ 0 & \frac{1}{2}e^{-t} \end{pmatrix}$$

Result from the classical approach

The classical approach yields a control: $u(t) = K(t)x(t)$ where $x(t)$ is the state of the system and $K(t) = \frac{g(t)}{f(t)} B_1^{-1}(t)$.
$K(t)$ is a $2X2$ matrix.
$K(t)$ is called the Full State Feedback gain matrix.
$K(t) = (K_{ij}(t))\ 1 \leq i \leq 2\ ,\ 1 \leq j \leq 2$ and

$$f(t) = \left(\frac{1+\sqrt{5}}{2} e^{\frac{1-\sqrt{5}}{2}(t-1)} - \frac{1-\sqrt{5}}{2} e^{\frac{1+\sqrt{5}}{2}(t-1)}\right)/\sqrt{5}$$

$$g(t) = -\left(e^{\frac{1-\sqrt{5}}{2}(t-1)} - e^{\frac{1+\sqrt{5}}{2}(t-1)}\right)/\sqrt{5}$$

The graph of the components K_{11} and K_{12} of $K(t)$

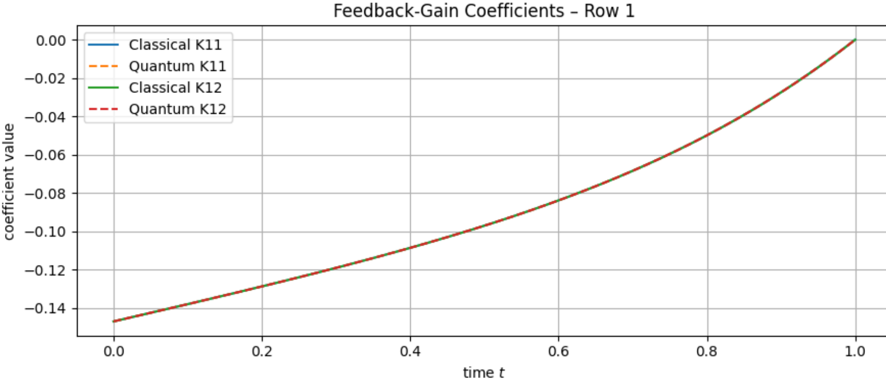

Figure: Comparison of classical and Enhanced hybrid quantum Full State Feedback gain matrix entries.

The graph of the components K_{21} and K_{22} of $K(t)$

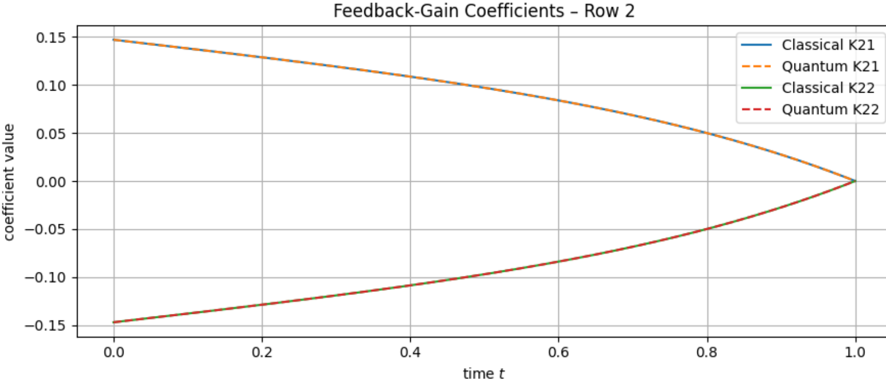

Figure: Comparison of classical and Enhanced hybrid quantum Full State Feedback gain matrix entries.

Conclusion

- We investigate the use of quantum computing to solve the time-varying Linear Quadratic Regulators (LQR) problem.
- We demonstrate an approach involving a change of variable that transforms a matrix Riccati equation into a set of second order matrix linear differential equations that can be solved on quantum computer using a form of HHL algorithm.
- We provide experimental results showing this approach yields results equivalent to classical calculations, but leverages quantum algorithms to achieve a faster, more computationally efficient method.

Question?