Feedback Control of Underactuated Systems using Time-Varying Linear Quadratic Regulator

Shiva Kumar Tekumatla

***Abstract*— This paper discusses the usage of time-varying linear quadratic regulator for underactuated systems. A two-wheeled cart-pole system (Segway) is used as a reference to study the performance of this control method and compare it with other control methods such as state feedback and LQR. Dynamic model for this system is designed using non-Holonomic constraints. A total of six states of this system are controlled using these control methods. An optimized trajectory is generated for all the states and inputs using optimized based control and is used for time-varying LQR. Overall, the better performance is observed with time varying LQR compared to other methods at various initial conditions. An attempt to simulate these control schemas is done on gazebo simulator as well.**

I. INTRODUCTION

There is no scarce for mobile robots that can navigate from one place to another. Numerous mechanisms have been studied over the years to establish an efficient mode of navigation. Out of all these modes, differential drive mobile robots remain to be the best navigators for most of the terrains. Though four-wheeled mobile robots perform best in most terrains, there have been design that use less than four wheels for navigation. A two wheeled pole cart system is a good example of such systems. A modern-day Segway is a pole-cart system that is been used by many for day-to-day transportation.

There have been many studies to implement different types of control methods to this system. Oryschuk *et al*. [1] LQR to stabilize the robot platform at upward configuration. Authors considered rotation of two wheels and pitch of the body as generalized coordinates which makes the size of state vector to be six. Finally, they used four states to stabilize the platform at upward configuration. Kim *et al*. [2] used state dependent Riccati equation (SDRE) to control two wheeled inverted pendulum mobile robot. Authors used straight movement, pitch and yaw as generalized coordinates and then tested performance of the SDRE on this system. They have also compared the results of SDRE with LQR. Acar *et al*. [3] used a two-wheeled mobile platform to implement manipulation. Authors used nonlinear backstepping method to control stability of passive joint. They used a candidate Lyapunov function as a variable of error and its integration and proved the system's stability.

Grandia *et al.* [4] provided nonlinear model predictive control for Segway using control Lyapunov functions. Authors presented unique and novel methods for unification of control Lyapunov functions and nonlinear model predictive control. They used SQP algorithm to efficiently solve nonlinear optimization problem occurred in this method. Gurriet *et al.* [5] tested a safety critical control framework that is designed based on active set invariance on two-wheeled mobile pendulum. Kim *et al*. [6] designed robust control method for Segway with unknown parameters and model uncertainties.

Figure 1 A commercial Segway 9bot S

Dynamic models of the Segway are studied well over the years and there have been numerous methods to obtain the dynamic model. The Segway is a mobile robot system with non-holonomic constraints. Because of these constraints, use of traditional methods such as Lagrangian is not straight forward. Once all the Lagrangian mechanics is obtained, all the nonholonomic constraints must be expressed in Pfaffian form. This can be used to with lagrange multipliers to obtain final equations of motion. This type of method is studied by Tuttle [7]. Author provided a detailed study on Segway dynamics. These dynamics are used for the study of time-varying LQR.

Time-varying LQR is a control method that is similar to LQR, but instead of constant gains as in LQR, Time-varying LQR has gains that change over the time based on states. This method of control requires optimized trajectories for both states as well as inputs. Direct-Multiple Shooting [8], Direct Collocation [9], Hermite-Simpson methods, and Pseudo Spectral methods are some of the methods that can be used to compute optimized trajectories for mechanical systems.

Implementation of Time-varying LQR to different under actuated systems was studied by Farzan *et al*. [10]. Authors developed optimized trajectories required [11] for this control using multiple-shooting and parametric trajectory approaches.

Even though there has been an extensive study of the various control methods on Segway, none of the authors have implemented time varying LQR on this system.

II. Dynamic Model

A. Lagrange Mechanics for nonholonomic conditions

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = M_q \ddot{q} - \beta_q \qquad (1)$$

The above equation represents the equation of motion of a dynamic system in terms of a positive mass matrix M and velocity matrix β. The subscript q represents the generalized coordinates. If the system has external forces acting on it, then the equation 1 can be written as equation 2.

$$M_q \ddot{q} - \beta_q = F_q \vec{u} \qquad (2)$$

Here F_q is the input coefficient matrix, and $F_q \vec{u}$ is generalized input forces. If a system has nonholonomic conditions, then these constraints must be explained on Pfaffian form as shown in equation 3.

$$H\dot{q} = 0 \qquad (3)$$

Here H is the matrix containing all constrains on the system. After including the constraints, the equation 2 will change to equation 4.

$$M_q \ddot{q} = \beta_q + H\vec{\lambda} + F_q \vec{u} \qquad (4)$$

λ is the vector containing all the lagrange multipliers of the system. These multiplies can be eliminated by multiplying both sided with null space of H. After this operation, the resulting equation is shown by equation 5.

$$M_p \ddot{p} = \beta_p + F_p \vec{u} \qquad (5)$$

The resulting equation shows the modified equations of motion for p generalized coordinates with $F_p \vec{u}$ generalized input forces.

B. Dynamics of Segway

A general Segway has three main parts, the body and two wheels. There are some constraints assumed for the Segway to obtain nonholonomic conditions. First assumption is that wheels do not lift off the ground. Second assumption is that the wheels always move perpendicular to the axle direction. The third condition is that the wheel do not slip on the ground. The Figure 2 shows the components of the Segway that is used for the dynamic model.

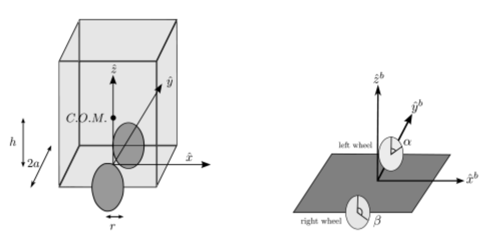

Figure 2 Segway Components

There are a total of six generalized coordinates for this Segway which are given by equation 6.

$$q = [x \;\; y \;\; \psi \;\; \theta \;\; \alpha \;\; \beta \; \dot{x} \; \dot{y} \; \dot{\psi} \; \dot{\theta} \; \dot{\alpha} \;\; \dot{\beta}] \qquad (6)$$

Using the afore mentioned nonholonomic conditions and choosing a clever null space matrix of Pfaffian matrix, these generalized coordinates can be changed to equation 7. Parameters used for this model is given by Table 1.

$$p = [l \;\; \psi \;\; \theta \;\; v \;\; \dot{\psi} \;\; \dot{\theta}] \qquad (7)$$

And the equations of motions are given by equation 5 and matrices M_p , β_p and F_p are as follows.

$$M_p = \begin{pmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{pmatrix} \qquad (8)$$

Where:

$$\mathrm{M11} = m\sin(\psi)^2 + \frac{2\,\mathrm{Jw}}{r^2} + m\cos(\psi)^2$$

$$\mathrm{M13} = \mathrm{M31} = h\,\mathrm{mb}\cos(\theta)\cos(\psi)^2 + h\,\mathrm{mb}\cos(\theta)\sin(\psi)^2$$

$$\mathrm{M22} = 2\,a^2\,\mathrm{mw} + \mathrm{Jz}\cos(\theta)^2 + \sin(\theta)^2\,(m\,h^2 + \mathrm{Jx}) + \frac{2\,\mathrm{Jw}\,a^2}{r^2}$$

$$\mathrm{M33} = \mathrm{mb}\,h^2 + \mathrm{Jy}$$

M12=M21=M23=M32=0

$$\beta_p = \begin{pmatrix} \beta 1 \\ \beta 2 \\ \beta 3 \end{pmatrix} \qquad (9)$$

Where:

$$\beta 1 = \cos(\psi)\left(h\,\mathrm{mb}\cos(\psi)\sin(\theta)\,\dot{\psi}^2 + 2\,h\,\mathrm{mb}\cos(\theta)\sin(\psi)\,\dot{\psi}\,\dot{\theta} + m\,v\sin(\psi)\,\dot{\psi} + h\,\mathrm{mb}\cos(\psi)\sin(\theta)\,\dot{\theta}^2\right)$$

$$\beta 2 = -v\left(h\,\mathrm{mb}\,\dot{\psi}\sin(\theta)\,\cos(\psi)^2 + h\,\mathrm{mb}\,\dot{\psi}\sin(\theta)\sin(\psi)^2\right) - \dot{\psi}\,\dot{\theta}\cos(\theta)\sin(\theta)\,(2\,\mathrm{mb}\,h^2 + 2\,\mathrm{Jx} - 2\,\mathrm{Jz})$$

$$\beta 3 = \cos(\theta)\sin(\theta)\,(\mathrm{mb}\,h^2 + \mathrm{Jx} - \mathrm{Jz})\,\dot{\psi}^2 + g\,h\,\mathrm{mb}\sin(\theta)$$

$$F_p = \begin{pmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{-a}{r} & \frac{a}{r} \\ -1 & -1 \end{pmatrix} \qquad (10)$$

Table 1Parameters used for dynamic modeling

Parameter	Symbol	Value
Mass of the body	mb	2.313 Kg
Mass of the wheel	mw	0.141 Kg
Radius of the wheel	r	0.0615 m
Half the width of Segway	a	0.165 m
MoI in X direction	Jx	0.1986 Kg-m^2
MoI in Y direction	Jy	0.1942 Kg-m^2
MoI in Z direction	Jz	0.0056 Kg-m^2
Center of mass	h	0.254 m

*MoI – Moment of Inertia

III. Control to Upward Configuration

The above defined dynamic model has many equilibrium points. This is because the variables l and ψ can be anything but the value of θ is either 0° or 90°. At the upward configuration, the value of θ is 0° . For testing various control methods, [0 0 0 0 0 0] is assumed to be the goal point. This point is also an equilibrium point. Jacobian linearization method is used to linearize the system around this point. The state space of the model around this point is computed and given by equation 11.

$$\dot{p} = A_{eq}p + B_{eq}u \qquad (11)$$

A_{eq} and B_{eq} are the state matrix and input matrix at the above equilibrium point. At this equilibrium point, the system is not stable but controllable. Hence, we can design state feedback, LQR and Time varying LQR control for this system.

A. State feedback control

To design a state feedback control, the systems eigen values must be placed such a way that the real part of the eigen values should be strictly negative. This is to make sure the closed loop state matrix is asymptotically stable. Figures 3 &4 show the system's performance at eigen values placed at $[-1+3i\,, -1-3i, -4, -2, -2+i, -2-i]$.

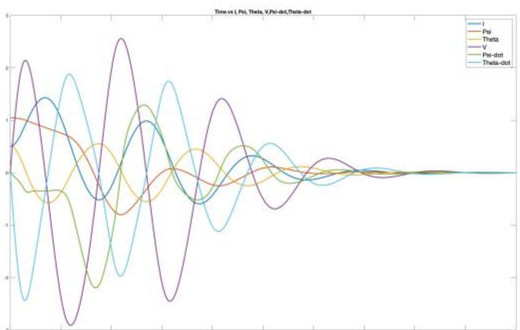

Figure 3 States of the system changing over time with state feedback controller

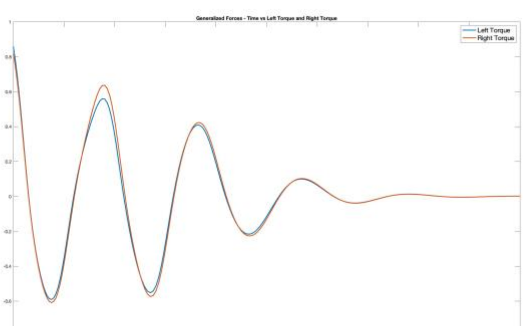

Figure 4 Inputs required for the system to reach upward configuration with state feedback controller

Different set of eigen values can be tried to achieve the best performance using state feedback controller. However, this method will not be able to give more control over the system, and it can be difficult to understand the relationship between desired eigenvalues and the resulting behavior of the system. However, using LQR we can focus on achieving good results with states and required inputs, and overall system's behavior is easy to understand.

B. Linear Quadratic Regulator

Using LQR, system's overall performance can be optimized at minimal cost. Each of the individual states can be given different weightage while minimizing the cost or to focus on control one certain states more than the other. The same way inputs can also be given different weightage. The cost optimization problem of LQR is given by equation 12.

$$J(p,u) = \int_{-\infty}^{\infty} (p(t)^T Q p(t) + u(t)^T\, R u(t))dt \qquad (11)$$

The gains required for a given set of Q and R matrices are computed by solving Algebraic Ricatti Equation.

Figures 5 & 6 shows the performance of LQR system with Q and R as identity matrices.

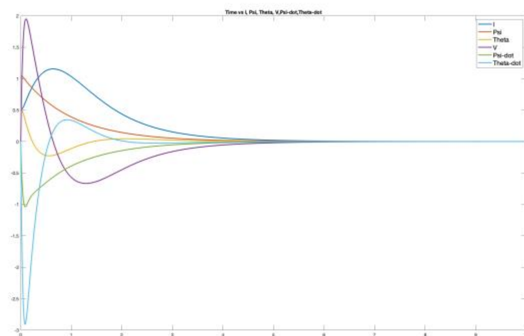

Figure 5 States of the System changing over time with LQR Controller

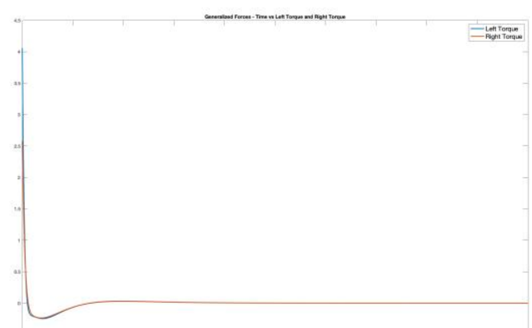

Figure 6 Inputs required over the time to reach upward configuration using LQR controller

C. Optimal Trajectory Generation

Though the performance with LQR is satisfactory, it can further be improved using time-varying LQR. In time-varying LQR the gains are not constant unlike in LQR. The same way gains are computed for LQR using linearized state and input matrices, time-varying gains computed if a trajectory is defined for the system. The system must be linearized around this trajectory and at each steps gains should be computed.

But generating trajectories is a complex task. Many of the methods such as Direct Multiple shooting, Direct colocation etc. can be used. However, the system's bounds, and input bounds must be taken into consideration with a good accuracy.

There are a good number of pre-developed modules available to do this task programmatically. A good definition of dynamics and the system bounds are given as input to this module and an optimized trajectories for system states and inputs are obtained. Control effort optimization problem for stage cost is defined below:

minimize: $J(u) = \sum_{k=1}^{n}(u_k{}^2)$

This optimization problem is subjected to $[0 \ \frac{\pi}{3} \ \frac{\pi}{6} \ 0 \ 0 \ 0]$ as initial conditions and [0 0 0 0 0 0] as final conditions. The same way boundary cost for the system is defined as the time (t_f) for the system to reach stability. The figure 7 shows the generated optimized trajectories.

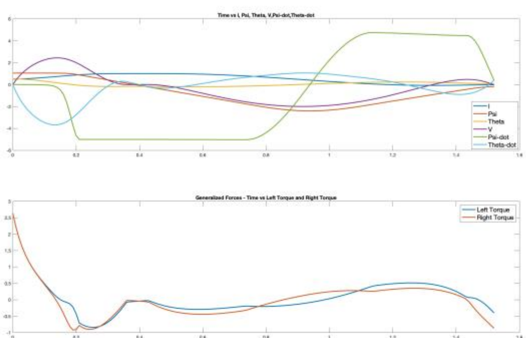

Figure 7 Optimized state and input trajectories

As seen from these plots, systems converged to the required states in a small timeframe and with smaller inputs compared to LQR. Figure 8 shows the system sates simulation with the generated input trajectories without using time varying LQR controller.

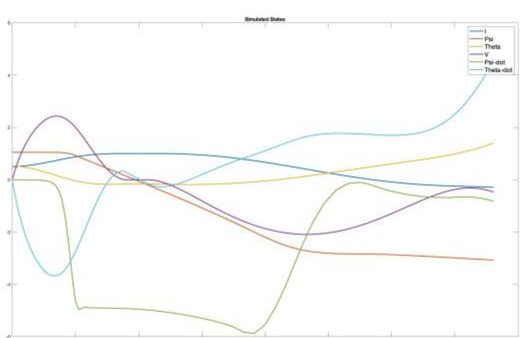

Figure 8 Simulated states with generated input trajectories

The system did not converge to desired trajectories with a great accuracy without using a closed loop controller.

D. *Time-Varying LQR*

The generated optimal trajectory is used with the state space model to find time-varying gains. The system is linearized around this trajectory at each point in time. To find the time varying gains, the differential Riccati equation given by equation 13 must be solved by integration backwards in time. The cost of the time-varying LQR is given by equation 12.

$$J(p,u) = \int_{-\infty}^{\infty} (\overline{p(t)}^T Q \overline{p(t)} + \overline{u(t)}^T R \overline{u(t)})dt + (\overline{p(t)}^T Q_f \overline{p(t)}$$

$$\overline{p(t)} = p(t) - p(t)_{ref} \text{ and } \overline{u(t)} = u(t) - u(t)_{ref}$$

$$\dot{P} = -(A^T P + PA - PBR^{-1}B^T P + Q) \qquad (13)$$

In this equation *A* and *B* are state and input matrices linearized around time varying state and input trajectories. Similar to LQR , Q, Q_f and R are constant matrices. The performance of time-varying LQR is shown in figure 9.

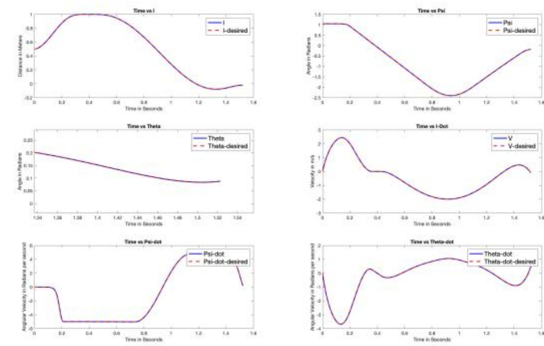

Figure 9 Actual states and desired states of the system with time varying LQR controller

The system converged to the desired trajectories right from the beginning and there was no error at all even at the final stage. But if the initial conditions are set to offset values, the system performance is not satisfactory as seen in figure 10.

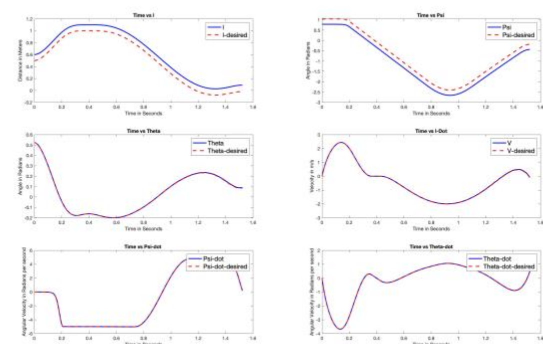

Figure 10Actual states and desired states of the system with time varying LQR controller with offset initial conditions

This can be due to an improper controller design. The better performance can be achieved by either varying the optimized trajectory parameters or the controller parameters. This part of the work is not presented in this paper.

IV. GAZEBO SIMULATION

A gazebo model for two wheeled cart-pole system is downloaded from GitHub [12]. This model has various sensors that can be subscribed to using ROS program. The model states and the sensors data used to compute the state vector in real-time and integrated with feedback controllers designed above. However, due to dynamics' discrepancies between gazebo and computed dynamics, the system performance in gazebo is not satisfactory.

V. CONCLUSIONS AND FUTURE WORK

In this paper, state feedback, LQR and time-varying LQR controllers are designed for a Segway, and the results are compared. Overall, the LQR controller performed better than state feedback controller and time-varying LQR performed better than the LQR controller. The dynamic model using non-holonomic constraints used to build these control schemas. Finally, an attempt to simulate the controller's performance on a gazebo model was done. However, the results with gazebo were not satisfactory. In the future, a study for better optimal trajectory generation and the study of system's behavior for various boundary conditions will be studied. The discrepancies between gazebo model and the dynamics will be explored and an simulation of these control methods will be attempted on the gazebo model.

REFERENCES

[1] P. Oryschuk, A. Salerno, A. M. Al-Husseini and J. Angeles, "Experimental Validation of an Underactuated Two-Wheeled Mobile Robot," in IEEE/ASME Transactions on Mechatronics, vol. 14, no. 2, pp. 252-257, April 2009, doi: 10.1109/TMECH.2008.2007482.

[2] S. Kim and S. Kwon, "Nonlinear Optimal Control Design for Underactuated Two-Wheeled Inverted Pendulum Mobile Platform," in IEEE/ASME Transactions on Mechatronics, vol. 22, no. 6, pp. 2803-2808, Dec. 2017, doi: 10.1109/TMECH.2017.2767085.

[3] C. Acar and T. Murakami, "Underactuated two-wheeled mobile manipulator control using nonlinear backstepping method," 2008 34th Annual Conference of IEEE Industrial Electronics, 2008, pp. 1680-1685, doi: 10.1109/IECON.2008.4758206.

[4] Grandia, R., Taylor, A., Singletary, A., Hutter, M., & Ames, A. (2020). Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions. Robotics: Science and Systems XVI. doi:10.15607/rss.2020.xvi.098.

[5] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron and A. Ames, "Towards a Framework for Realizable Safety Critical Control through Active Set Invariance," 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), 2018, pp. 98-106, doi: 10.1109/ICCPS.2018.00018.

[6] Kim BW, Park BS. Robust Control for the Segway with Unknown Control Coefficient and Model Uncertainties. Sensors (Basel). 2016 Jun 29;16(7):1000. doi: 10.3390/s16071000. PMID: 27367696; PMCID: PMC4970050.

[7] Tuttle, J.. "Studies of systems with nonholonomic constraints: The Segway and the Chaplygin sleigh." (2014).

[8] Bock H.G., Diehl M.M., Leineweber D.B., Schlöder J.P. (2000) A Direct Multiple Shooting Method for Real-Time Optimization of Nonlinear DAE Processes. In: Allgöwer F., Zheng A. (eds) Nonlinear Model Predictive Control. Progress in Systems and Control Theory, vol 26. Birkhäuser, Basel. https://doi.org/10.1007/978-3-0348-8407-5_14

[9] Von Stryk,O.(1999) User's Guide for DIRCOl—A direct Collocation Method for the Numerical Solution of Optimcal Problems.

[10] S. Farzan, A. Hu, E. Davies and J. Rogers, "Feedback Motion Planning and Control of Brachiating Robots Traversing Flexible Cables," 2019 American Control Conference (ACC), 2019, pp. 1323-1329, doi: 10.23919/ACC.2019.8814894.

[11] S. Farzan, A. Hu, E. Davies and J. Rogers, "Modeling and Control of Brachiating Robots Traversing Flexible Cables," 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 1645-1652, doi: 10.1109/ICRA.2018.8461036.

[12] Robustify Github *Self-balancing robot simulation model for ROS / Gazebo* https://github.com/robustify/teeterbot