الجمهورية الجزائرية الديمقراطية الشعبية

Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of higher education and scientific research

جامعة غرداية

Order's N°

University Of Ghardaia

كلية العلوم والتكنولوجيا

Sciences and technologies faculty

قسم الألية والأنظمة

Department of sciences and technology

A Thesis submitted in partial fulfillment of requirements for the degree of

MASTER

Domaine: Sciences and technology

Field: Automation

Speciality: Automatic and systems

Topic

GENETIC ALGORITHM IMPLEMENTATION FOR OPTIMIZING LINEAR QUADRATIC REGULATOR

Presented by:

Dahma Marwa
Ben Khelifa Sarah

Date 02/06/2022

In front of the jury:

Bechaouat Mohcene	**M.C.A**	Univ. Ghardaïa	Supervisor
Benchaabae Achour	**M.A.A**	Univ. GhardaïA	Examiner
Bekkar Belgacem	**M.C.B**	Univ. Ghardaïa	Examiner

Academic year 2021/2022

Acknowledgment

First of all, we thank God who gave us the strength and the patience to accomplish this modest work.

We express our deepest gratitude to our supervisors Dr MOHCEN BECHOUAT, who agreed to supervise this research and who provided us with wise guidance and valuable feedback for the completion of this study.

Moreover, we want to thank everyone who provided us with sufficient information and notes that helped us realize our project.

Finally, we would like to thank all the people who have helped us encouraged & supported us directly and indirectly we express our sincere gratitude to all of them.

Dedication

Dedication

I dedicate this modest work to my pure father soul god bless him and the great mother who was all her priority.to my only refuge Fatiha.

This work it dedicated to every person who helped me, even if with a word, for everyone who motivated me from the beginning until the day of my graduation, for every person who contributed to raising my moral. For every person who told me you can do it, for family, friends and cousins.

For my lovely partner Marwa . She’s always been my support.

For my childhood friend Marwa, who helped me a lot.

For me, to the hope, love, life, and passion.

I dedicate this research to every student who open it and read it, I hope you find what you want. And the most important thing that this work will become an ongoing charity for us.

Ben. Sarah

Dedication

I dedicate this modest work to my parents and my sisters Nacira and Safa especially **my mother**, the patient woman who give us love and care and believed in me and supported me.

For my partner and my sister Sarah, I tell her we finally did it.

For my uncle Mustapha who supported me and helped me a lot, I thank hem

For our colleague Fatima for her patience, who worked under our pressure, and I thank her for standing by us during our journey together.

For all my teachers who taught me from elementary school until now, and I thank them for their efforts with us.I dedicate this modest work for everyone who helped me in it, even if only a little

Dah. Maroua

Résumé

Le travail de ce mémoire porte la modélisation et la synthèse de lois de commandes linéaires pour un pendule inversé rotatif. Ce système non-linéaire caractérisé par un dynamique instable, ce dernier représente un système intéressent dans le domaine de l'industriel du fait de sa difficulté, qui constitue une matière riche pour étudier et expérimenter les dernières lois de commandes, les lois maintiennent le pendule en équilibre vers le haut après que le pendule a été placé manuellement autour de sa position d'équilibre instable. Nous présentons d'abord un modèle mathématique non-linéaire sous la forme d'équation différentielles et sous la forme d'une représentation d'état. Ensuite, nous avons présenté une linéarisation du modèle obtenu autour du point d'équilibre instable. Finalement, nous avons synthétisé une des commandes linéarises, linéaire quadratique régulateur (LQR) et linéaire optimisé par les algorithmes génétiques.

MOTS CLES : Système non linéaire, Pendule inversé rotatif, Commandes linéaires, modélisation mathématique, les algorithmes génétiques.

ملخص

يركز عملنا هذا على نمذجة و توليف قوانين التحكم الخطي للنواس الدوار المقلوب .يتميز هذا النواس الغير خطي بديناميكية غير مسقرة ,ويمثل هذا الأخير نظاما مثيرا للإهتمام في المجال الصناعي لصعوبته مما يشكل مادة دسمة لدراسة وتجربة احدث قوانين التحكم .الغرض من هذه القوانين الحفاظ على النواس متوازنا إلى الأعلى بعدما يتم وضع النواس يدويا حول موضع توازنه الغير مستقر .أولا قدمنا نمودجا رياضيا غير خطي في شكل معادلات تفاضلية وفي شكل ثمتيل حالة .قدمنا لاحقا نمودجا خطيا تم الحصول عليه حول نقطة التوازن الغير مستقرة وأخيرا قمنا بتطبيق أحد عناصر التحكم الخطية وهي المنظم الخطي التربيعي المحسن بالخوارزميات الجينية .

<u>كلمات المفتاحية:</u> النظام غير الخطي، النواس الدوار المقلوب، أدوات التحكم الخطية، النمذجة الرياضية ،الخوارزميات الجينية .

Abstract

This thesis focuses on the modeling and synthesis of linear control for the rotary inverted pendulum available on our laboratory. With its unstable and highly nonlinear dynamics, the rotating inverted pendulum represents a very interesting system to study, and is an attractive one for testing new control laws. The goal of these controls is to maintain the pendulum at its high equilibrium position when the pendulum is manually placed around its unstable equilibrium position. In the first step, we have given a non-linear mathematical model in the form of differential equations and in the form of a state representation. We then presented a linearization of the model obtained around the unstable equilibrium point. Finally, we have synthesized a linearized control, namely, linear quadratic regulator (LQR) and the optimized linear quadratic regulator using genetic algorithms.

Key-Words: Nonlinear system, Inverted rotating pendulum, Linear controls, mathematical modeling, genetic algorithms.

Contents

List of figures

List of tables

List of abbreviations

LQR: Linear quadratic regulator

GAs: Genetic algorithms

RIP: Rotary inverted pendulum

Introduction

Automation is the art of modeling and analyzing the controller systems it is also the one that identifies dynamic systems for processing information and making decision to minimize human intervention to accomplish difficult missions that require effort beyond human and physical capabilities.

The inverted pendulum among all the pendulums has an important position in the industry as a highly unstable nonlinear system. This system represents an ideal reference for studying and testing the latest control methods.

The rotary inverted pendulum also, called the Furuta pendulum, consists of an arm in the horizontal plane which is attached by a direct current motor, at the end of this horizontal arm is connected another arm that rotates freely in the vertical plane using a controller.

The goal of this work is to ensure that the pendulum stabilizes around its point of unstable equilibrium. To reach and get a satisfying result we use a Linear Quadratic Regulator (LQR) command.

This study is organized as follows:

In the first chapter we study the modeling on the rotary inverted pendulum. Where we introduce the rotary inverted pendulum and its components, the different pendulum types, and its real-time application. Moreover, we elaborate on its dynamic model using Eluer-lagrang method. Eventually, we will be presenting this system's state space.

In The second, chapter we have presented in general the optimal control and we discussed the fundamental notions of linear quadratic regulator and some basic concepts, then we have exposed general definition on genetic algorithms.

The third chapter is a simulation part where we apply the two commands on our system after we have illustrated the results obtained.

And finally, in the fourth chapter we implemented the three control that we synthesized before, on the Quanser QUBE-Servo model which is an inverted pendulum model

Rotary, in order to test the performance and robustness of the three controls.

CHAPTER I

Modeling of the Rotary inverted pendulum

Chapter 1: Modelling of the rotary inverted pendulum

1.1 Introduction

In this chapter, we study the modeling of the rotary inverted pendulum. We introduce the rotary inverted pendulum and its components, the different pendulum types, and its real-world application. Moreover, we elaborate on its dynamic model using the Eluer-lagrang method. Eventually, we will be presenting this system's state space.

1.2 The inverted pendulum

An inverted pendulum is a rod placed in a position of unstable equilibrium (180° vertical) on a base that is either fixed or mobile, somewhat similar to balancing a rod on the fingers; where we have to constantly adjust our hands position to stabilize the rod. The Inverted pendulum system represents a significant class of nonlinear under-actuated mechanical systems that exhibit numerous problems present in industrial applications, such various external disturbances or nonlinear behaviours under different operation conditions.

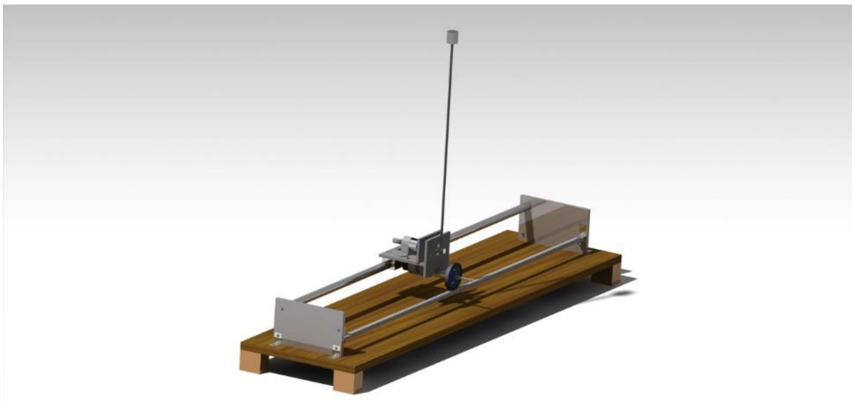


Figure 1-1 : Illustrative example of inverted pendulum

1.3 The different inverted pendulum types:

There exist different types of the inverted pendulum and below are some examples:

1.3.1 Classic inverted pendulum

This mechanical system is a cart with mass m that can move horizontally and freely on a guide rail by supporting a rod with mass M [10]. This rod can rotate freely around a pivot.

A force f applied to cause the cart to move and the deviation of the pendulum by an angle θ in accordance to the vertical.

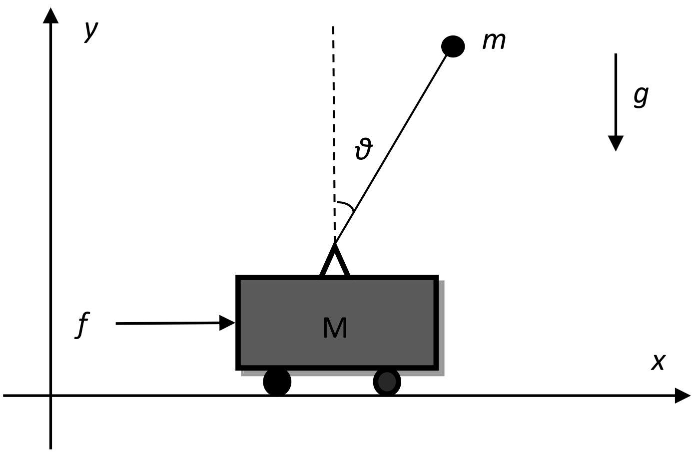


Figure 1-2 classic inverted pendulum

1.3.2 The inverted double pendulum

For this type, there are two different architectures:

a) The cascade architecture:

It has the same principal as the classic pendulum, the only difference being that is has two freely rotating rods, one that rotate around the pivot with an angle θ1(t) and the other with an angle θ2(t) around the second joint between the rods [11].

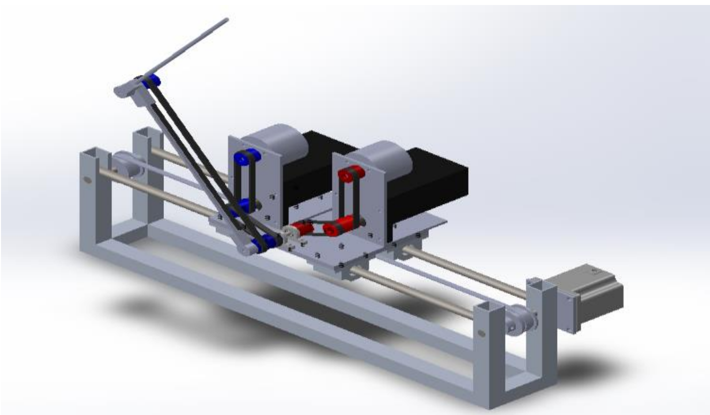


Figure 1-3 A cascading inverted double pendulum.

b) ***Parallel architecture:***

In this architecture the cart supports two independent rods; a rod L of length l_1 and another rod B of length l_B both that can freely rotate. The displacement $x_{(t)}$ of the cart will cause a deviation of an angle θ_L from the vertical of the first rod and the angle θ_B accordance to the vertical on the second rod [12].

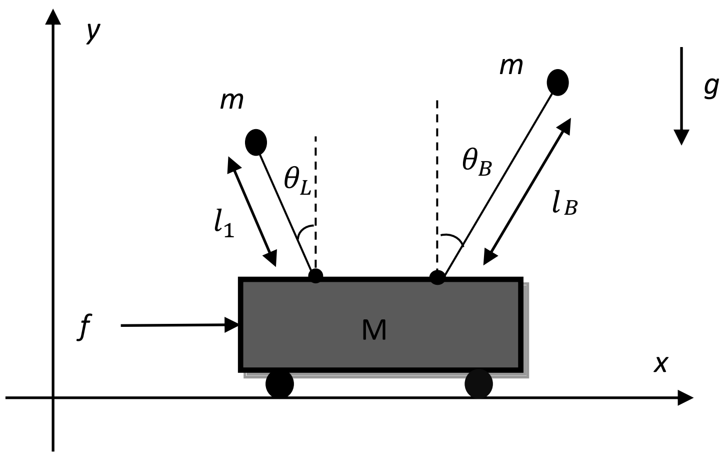


Figure 1-4 A double parallel inverted pendulum

1.3.3 Inverted pendulum stabilized by flywheel

The operating principle of this system is based on the rotational movement of the flywheel which is caused by the dynamic effects that it induces. The pendulum rotation is a system composed of two mechanical bodies: an inverted pendulum in free rotation around a pivot linked to the frame and an actuated flywheel whose center of mass coincides with the end of the pendulum [9] .

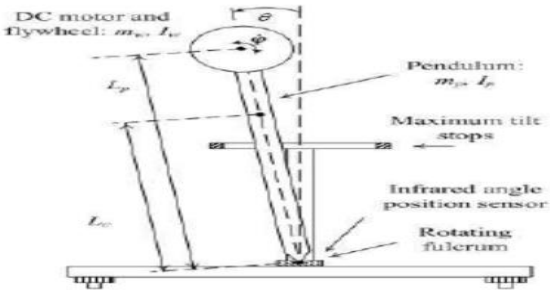


Figure 1-5 Inverted pendulum stabilized by flywheel [9]

1.3.4 Inverted pendulum on two wheels

this type of pendulum consists of a mobile base (the axle and the two wheels) surmounted by a rotating inverted pendulum free around a pivot (passive articulation) between the axle and the pendulum rod. The corner tilt of the pendulum with respect to the vertical is denoted ψ.

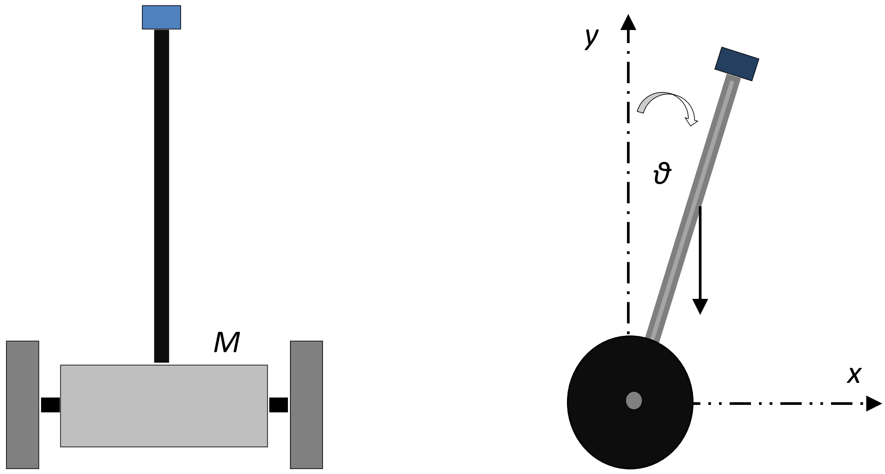


Figure 1-6 Inverted pendulum on two wheels

1.3.5 Rotary Inverted pendulum

The rotary inverted pendulum is composed of an arm actuated in rotation in the horizontal plane, at its end is added an inverted pendulum mounted in unstable equilibrium. The infinite rotation of the arm ensures the stabilization and maintenance of the pendulum around the vertical at the point unstable balance [28] .

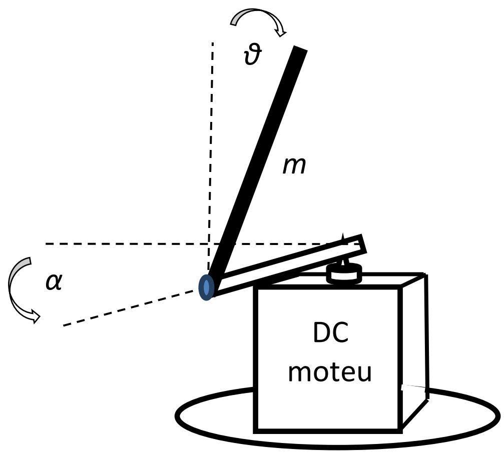


Figure 1-7 rotary Inverted pendulum [8]

1.4 Application of inverted pendulum

the inverted pendulum is used in several important fields and applications like

1.4.1 Medical field

IBOT: an automated electric wheelchair, which works based on the principle of the inverted pendulum. Equipped with four driving wheels It allows users to displacement on different types easily, This IBOT able to help people with mobility problems.

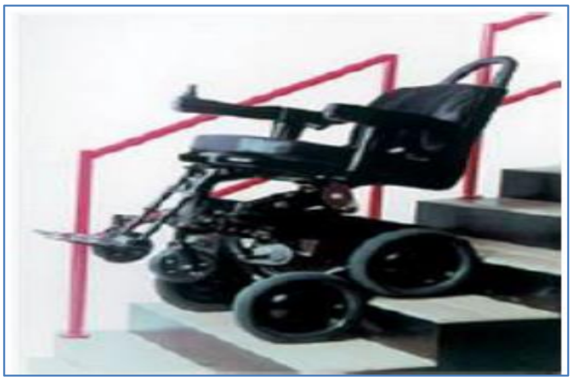


Figure 1-8 IBOT [7]

1.4.2 Transport field

It is a classical motorcycle which is based on the principle of the inverted pendulum. This motorcycle creates a self-balancing in the critical phases.

Figure 1-9 Honda Riding Assist [6]

1.4.3 In the aerospace:

the study of pendulum systems is of great importance, for example to control and stabilize the attitude of the satellite, the launch of rockets ... etc. Rocket works on the principle of pendulum inverted, which is a spacecraft, aircraft, vehicle or projectile that obtains thrust from a rocket engine. Rocket engine exhaust is formed entirely from propellant carried within the rocket. Rocket engines work by action and reaction and push rockets forward simply by expelling their exhaust in the opposite direction at high speed, and can therefore work in the vacuum of space.

1.4.4 In robotics:

Self-balancing Robot

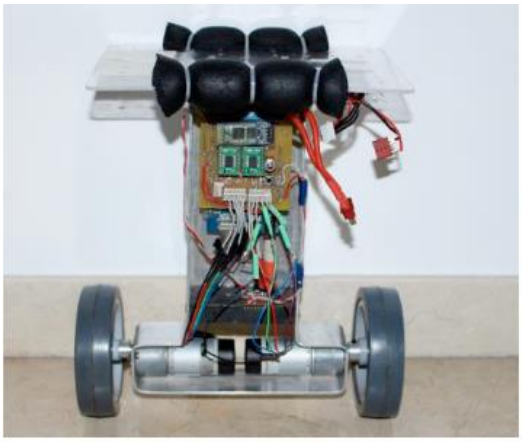


Figure 1-10   Self balancing Robot [5].

And as a successful application of the principle of the inverted pendulum we can mention

	
a) Segway	b) hoverboard

Figure 1-11 the application examples [6].

1.5 Dynamic modeling

The inverted rotary pendulum model is a combination of an arm that attaches the SRV02 system and a servo rotary base unite should turn in the CCW direction when the control voltage is positive $V_m > 0$. The pendulum is connected at the end of the rotary' s arm, more specifically at the arm's metal shafts as shown in

Figure 1-12 .

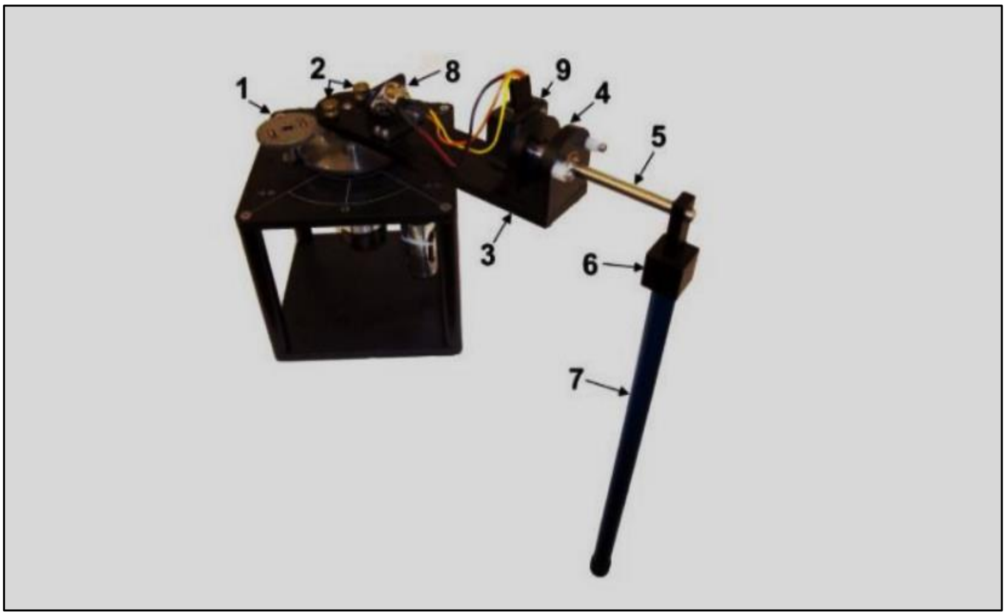


Figure 1-12 components of an inverted rotary pendulum [1].

N°	Components
1	SRV02
2	VIS
3	Arm
4	Shaft housing
5	Shaft
6	connection
7	Pendulum
8	Encoder connector
9	Encoder

Table 1-1 Nomenclature of the components of the rotary pendulum dynamic modeling

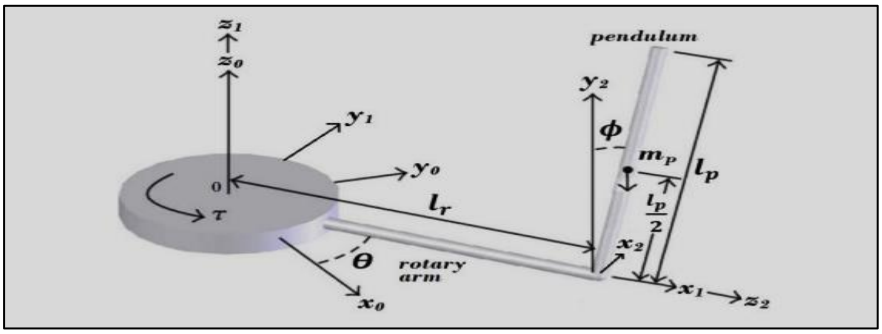


Figure 1-13 Rotary inverted pendulum and its coordinate systems [1].

1.6 Analytical modeling

In this section, dynamic modeling of RIP is performed by using two approaches,

the first approach is Newton's fundamental law of dynamics which is based on the concept of force, in the second approach, which is analytical in nature, the Euler-Lagrange is used to develop the equation of motion which is based on the principal conservation of mechanical energy.

In this work we used the Euler-Lagrange method to find the equation of motion.

The Lagrange function L is defined as the difference of the kinetic energy and potential energy, and the equations describing these two energies respectively are expressed in terms of the variables in the specified coordinate systems as shown in

Figure 1-12 .

$$L = T - V \tag{1-1}$$

kinetic energy of the system (T_{sys}):

$$T_{sys} = T_p + T_r \tag{1-2}$$

$$T_{sys} = \frac{1}{2} m_p V^2 + \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} J_r \dot{\theta}^2 \tag{1-3}$$

Potential energy of the system (T_{sys}):

$$V_{sys} = V_p + V_r \tag{1-4}$$

$$V_{sys} = \frac{1}{2} m_p g L_p cos\ \alpha \tag{1-5}$$

For reach the center speed of pendulum we need to determine the coordinate P_f of the pendulum center mass by:

$$P_x = L_r cos\ (\theta) + \frac{1}{2} L_p sin\ (\alpha) sin\ (\theta) P_y = L_r cos\ (\theta) - \frac{1}{2} L_p sin\ (\alpha) sin\ (\theta) P_z = \frac{1}{2} L_p cos\ (\alpha) \tag{1-6}$$

We take the derivatives of this coordinates to find the center speed of the pendulum

$$P^{\cdot}{}_x = -L_r \theta^{\cdot} sin\ \theta + \frac{1}{2} L_p (\alpha^{\cdot} cos\ \alpha sin\ \theta + \theta^{\cdot} cos\ \theta sin\ \alpha)\ P^{\cdot}{}_y = L_r \theta^{\cdot} sin\ \theta - \frac{1}{2} L_p (\alpha^{\cdot} cos\ \alpha sin\ \theta + \theta^{\cdot} cos\ \theta sin\ \alpha)\ P^{\cdot}{}_z = -\frac{1}{2} L_p \alpha^{\cdot} sin\ \alpha \tag{1-7}$$

We have:

$$P^{\cdot 2} = P^{\cdot 2}_{x} + P^{\cdot 2}_{y} + P^{\cdot 2}_{z} \tag{1-8}$$

So, by replacing in () we find:

$$E_c = \frac{1}{2}\left(J_r + m_p L_r^2 + \frac{1}{4}\, m_p L_p^2 sin^2\ \alpha \right) \theta^{\cdot 2} + \frac{1}{2}\left(J_p + \frac{1}{4}\, m_p L_p^2 \right) \alpha^{\cdot 2} - \frac{1}{2}\, m_p L_r L_p cos\ \alpha \theta^{\cdot} \alpha^{\cdot} \tag{1-9}$$

Then L becomes:

$$L = \frac{1}{2}\left(J_r + m_p L_r^2 + \frac{1}{4}\, m_p L_p^2 sin^2\ \alpha \right) \theta^{\cdot 2} + \frac{1}{2}\left(J_p + \frac{1}{4}\, m\,_p L_p^2 \right) \alpha^{\cdot 2} - \frac{1}{2}\, m_p L_r L_p cos\ \alpha \theta^{\cdot} \alpha^{\cdot} - \frac{1}{2}\, m_p g L_p cos\ \alpha \tag{1-10}$$

1.7 The Euler-Lagrange method:

The Euler–Lagrange equation is mathematical equation that was developed in the 1750s by Leonard-Euler and joseph louis- Lagrange in connection with their studies of the autochrome problem. that result plays a fundamental role in the calculus of variations. We find this equation in many real problems of arc length minimization, such as the brachistochrone problem or even geodesic problems Specifically, the equations that describe the motion of the rotating arm and the pendulum with respect to the voltage of the servomotor will be obtained using of equation of Euler-Lagrange:

$$\frac{d}{dt}\left(\frac{dL}{dx^{\cdot}}\right) - \frac{dL}{dx} = Q \tag{1-11}$$

With our choice of generalized displacement vector $q(t)T = [\theta(t)\alpha(t)]$

$$\{\frac{d}{dt}\left(\frac{dL}{d\theta^{\cdot}}\right) - \frac{dL}{d\theta} = Q_1\ \frac{d}{dt}\left(\frac{dL}{d\alpha^{\cdot}}\right) - \frac{dL}{d\alpha} = Q_2 \tag{1-12}$$

the Euler-Lagrange equation can be written:

where, L = K - P and T represents torque applied by motor at rotary arm.

$$\{Q_1 = \tau - B_r \theta^{\cdot}\ Q_2 = -B_p \alpha^{\cdot} \tag{1-13}$$

With the torque τ, which generated by a servo motor is described by the equation:

$$\tau = \frac{\eta_g \eta_m K_g K_t \left(V_m - K_g K_m \theta^{\cdot} \right)}{R_m} \tag{1-14}$$

After solving the equations and linearizing them about operating point the following equations of motion are obtained:

$$\theta^{\cdot\cdot}\left(J_r + m_pL_r^2 + \frac{1}{4}m_pL_p^2 sin^2\right) + \alpha^{\cdot\cdot}\left(-\frac{1}{2}m_pL_pL_r cos\ \alpha\right) + \alpha^{\cdot}\theta^{\cdot}\left(\frac{1}{2}m_pL_p^2 sin\ \alpha cos\ \alpha\right) + \alpha^{\cdot\cdot 2}\left(\frac{1}{2}m_pL_rL_p sin\ \alpha\right) = Q_1 \theta^{\cdot\cdot}\left(-\frac{1}{2}m_pL_rL_P cos\ \alpha\right) + \alpha^{\cdot\cdot}\left(J_P + \frac{1}{4}m_pL_p^2\right) + \theta^{\cdot 2}\left(-\frac{1}{4}m_pL_p^2 sin\ \alpha cos\ \alpha\right) - \frac{1}{2}m_pgL_P sin\ \alpha = Q_2 \tag{1-15}$$

the motor and rotary inverted pendulum parameters are given with them units in Tableau 1-2

Symbol	Description	units	Values
Lp	Length of the pendulum	m	0.129
Lr	Length of the rotary arm	m	0.085
mp	Mass of pendulum	kg	0.024
J	pendulum Inertia	kgm^2	3.3xl0-5
Jr	Rotary arm Inertia	kgm^2	5.7xl0-5
Dr	Viscous damping coefficient	$\frac{Nms}{rad}$	0.0015
Dp	Pendulum damping coefficient	$\frac{Nms}{rad}$	0.0005

Table 1-3 parameters of the rotary inverted pendulum

1.8 Linearizing:

Linearizing equation (1-16) and (1-16) are linearized around the point of working $[\alpha, \theta, {}^{\cdot}, \alpha, {}^{\cdot}\ \theta, {}^{\cdot\cdot}\alpha^{\cdot\cdot}]$ the resulting linear equation of the inverted pendulum are defined as:

$$(m_pL_r^2 + J_r)\theta^{\cdot\cdot} - \frac{1}{2}\, m_pL_pL_r\alpha^{\cdot\cdot} = \tau - B_r\theta^{\cdot}\frac{1}{2}\, m_pL_rL_p\theta^{\cdot\cdot} + \left(J_p + \frac{1}{4}\, m_pL_p^2\right)\alpha^{\cdot\cdot} + \frac{1}{2}\, m_pL_pg\alpha = -B_p\alpha^{\cdot} \tag{1-16}$$

The linear stat space equation is:

Nonlinear rotary inverted pendulum equation, equation () the initial condition for all variables are zero means: $\alpha_0 = 0, \theta^{\cdot}{}_0 = 0, \alpha^{\cdot}{}_0 = 0, {}^{\cdot}$

$$f(z) = \left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 sin\,(\alpha)^2 + J_r\right)\theta^{\cdot\cdot} - \left(\frac{1}{2} m_p L_p L_r cos\,(\alpha)\right)\alpha^{\cdot\cdot} + \left(\frac{1}{2} m_p L_p^2 sin\,(\alpha) cos\,(\alpha)\right)\theta^{\cdot}\alpha^{\cdot} + \left(\frac{1}{2} m_p L_p L_r sin\,(\alpha)\right)\alpha^{\cdot 2} = \tau \tag{1-17}$$

Nonlinear rotary inverted pendulum equation, equation () the initial condition for all variables is zero:

The linearization of $f(z)$ relative to $\theta^{\cdot\cdot}$gives:

$$\left(\frac{\partial f(z)}{\partial \theta^{\cdot\cdot}}\right)|_{z=z_0} = m_p L_r^2 + \frac{1}{4} m_p L_p^2 sin\,(0)^2 + J_r = m_p L_r^2 + J_r \tag{1-18}$$

By linearizing $f(z)$ relative to $\alpha^{\cdot\cdot}$ we obtain:

$$\left(\frac{\partial f(z)}{\partial \theta^{\cdot\cdot}}\right)|_{z=z_0} = \frac{1}{2} m_p L_p L_r cos\,(0) = \frac{1}{2} m_p L_p L_r \tag{1-19}$$

All other terms are:

$$\left(\frac{\partial f(z)}{\partial \theta}\right)|_{z=z_0} = 0 \qquad \left(\frac{\partial f(z)}{\partial \alpha^{\cdot}}\right)|_{z=z_0} = 0 \tag{1-20}$$

$$\left(\frac{\partial f(z)}{\partial \theta}\right)|_{z=z_0} = 0 \qquad \left(\frac{\partial f(z)}{\partial \alpha}\right)|_{z=z_0} = 0 \qquad f(0) = 0 \tag{1-21}$$

$$f_{lin}(z) = f(0) + \left(\frac{\partial f(z)}{\partial \theta^{\cdot\cdot}}\right)|_{z=z_0}\theta^{\cdot\cdot} + \left(\frac{\partial f(z)}{\partial \alpha^{\cdot\cdot}}\right)|_{z=z_0}\alpha^{\cdot\cdot} + \left(\frac{\partial f(z)}{\partial \theta^{\cdot}}\right)|_{z=z_0} + \left(\frac{\partial f(z)}{\partial \dot{\alpha}}\right)|_{z=z_0}\dot{\alpha} + \left(\frac{\partial f(z)}{\partial \theta}\right)|_{z=z_0}\theta + \left(\frac{\partial f(z)}{\partial \alpha}\right)|_{z=z_0}\alpha \tag{1-22}$$

By evaluating equation, we obtain:

$$f_{lin}(z) = \left(m_p L_r^2 + J_r\right)\theta^{\cdot\cdot} - \frac{1}{2} m_p L_p L_r \alpha^{\cdot\cdot} \tag{1-23}$$

Integrating this into the original equation. we get the following equation of linear motion:

$$\left(m_p L_r^2 + J_r\right)\theta^{\cdot\cdot} - \frac{1}{2} m_p L_p L_r \alpha^{\cdot\cdot} = \tau - B_r\theta^{\cdot} \tag{1-24}$$

Linearizing the second equation of the nonlinear rotary inverted pendulum equation with initial condition: $\theta_0 = 0, \alpha_0 = 0, \theta^{\cdot}{}_0 = 0$, $\alpha^{\cdot}{}_0$=0

The same principles as those used to linearize the first equation of nonlinear motion can be used for this purpose. The left side of the equation is:

$$f(z) - \frac{1}{2} m_p L_p L_r \cos\cos(\alpha)\, \theta^{\cdot\cdot} + \left(J_p + \frac{1}{4} m_p L_p^2\right) \alpha^{\cdot\cdot} - \frac{1}{4} m_p L_p^2 \cos\cos(\alpha)\ \sin\sin(\alpha)\, \theta^{\cdot 2} - \frac{1}{2} m_p L_p g \sin(\alpha) \tag{1-25}$$

The linearization given in equation is used for this equation:

The solution to derivatives based on: $\theta, {}^{\cdot\cdot}\alpha\, {}^{\cdot\cdot} and\ \ \alpha$ is:

$$\left(\frac{\partial f(z)}{\partial \theta^{\cdot\cdot}}\right)|_{z=z_0} = -\frac{1}{2} m_p L_p L_r \tag{1-26}$$

$$\left(\frac{\partial f(z)}{\partial \alpha^{\cdot\cdot}}\right)|_{z=z_0} = J_p + \frac{1}{4} m_p L_p^2 \tag{1-27}$$

$$\left(\frac{\partial f(z)}{\partial \alpha}\right)|_{z=z_0} = -\frac{1}{2} m_p L_p g \tag{1-28}$$

The other derivatives based on are zero and $f(z_0) = 0$ evaluating the function $f_{lim}(z)$, we obtain:

$$f_{lin}(z) = -\frac{1}{2} m_p L_p L_r \theta^{\cdot\cdot} + J_p + \left(\frac{1}{4} m_p L_p^2\right) \alpha^{\cdot\cdot} - \frac{1}{2} m_p L_p g \alpha \tag{1-29}$$

$$-\frac{1}{2} m_p L_p L_r \theta^{\cdot\cdot} + J_p + \left(\frac{1}{4} m_p L_p^2\right) \alpha^{\cdot\cdot} - \frac{1}{2} m_p L_p g \alpha = -B_p \alpha^{\cdot} \tag{1-30}$$

Linear model in state space:

The linear state-space equations are:

$$\dot{x} = Ax + Bu \tag{1-31}$$

$$y = Cx + Du$$

Either x is the state, or is the control input. A, B, C, D are the state matrices. For the rotating inverted pendulum system, the state and output equations are defined as follows:

$$x^T = [\theta\ \alpha\ \dot{\theta}\ \dot{\alpha}], \qquad \dot{x}^T = [\dot{\theta}\ \dot{\alpha}\ \ddot{\theta}\ \ddot{\alpha}] \tag{1-32}$$

$$y^T = [\theta\ \alpha]$$

From the generalized definition of the coordinates in (1-12) and the linear equations Equation (1-17) and Equation (1-14), the matrix becomes:

$$\left[m_p L_r^2 + J_r\ \ -\frac{1}{2} m_p L_p L_r\ \ -\frac{1}{2} m_p L_p L_r\ J_p + \frac{1}{4} m_p L_p^2\right] [\ddot{\theta}\ \ddot{\alpha}\ ] + [B_r\ 0\ 0\ B_p\ ][\dot{\theta}\ \dot{\alpha}\ ] + \left[0\ -\frac{1}{2} m_p L_p g \alpha\right] = [\tau\ 0\ ] \tag{1-33}$$

Rearrange the matrix (1-18) to obtain:

$$\left[m_pL_r^2 + J_r \;\; -\frac{1}{2}m_pL_pL_r \;\; -\frac{1}{2}m_pL_pL_r \;\; J_p + \frac{1}{4}m_pL_p^2\right]\left[\ddot{\theta}\;\ddot{\alpha}\right] = \left[\tau - B_r\dot{\theta} \;\; \frac{1}{2}m_pL_pg\alpha - B_p\dot{\alpha}\right] \tag{1-34}$$

$$\left[m_pL_r^2 + J_r \;\; -\frac{1}{2}m_pL_pL_r \;\; -\frac{1}{2}m_pL_pL_r \;\; J_p + \frac{1}{4}m_pL_p^2\right]^{-1} = \frac{1}{J_T}\left[J_p + \frac{1}{4}m_pL_p^2 \;\; \frac{1}{2}m_pL_pL_r \;\; \frac{1}{2}m_pL_pL_r \;\; m_pL_r^2 + J_r\right] \tag{1-35}$$

The determinant of the matrix is equal to:

$$J_T = \left(m_pL_r^2 + J_r\right)\left(J_p + \frac{1}{4}m_pL_p^2\right) - \frac{1}{4}m_p^2L_p^2L_r^2 = J_pm_pL_r^2 + J_rJ_p + \frac{1}{4}J_rm_pL_p^2 \tag{1-36}$$

Resolve Acceleration Conditions

$$\left[\ddot{\theta}\;\ddot{\alpha}\right] = \frac{1}{J_T}\left[J_p + \frac{1}{4}m_pL_p^2 \;\; \frac{1}{2}m_pL_pL_r \;\; \frac{1}{2}m_pL_pL_r \;\; m_pL_r^2 + J_r\right]\left[\tau - B_r\dot{\theta} \;\; \frac{1}{2}m_pL_pg\alpha - B_p\dot{\alpha}\right] \tag{1-37}$$

From matrix multiplication, the first equation is:

$$\ddot{\theta} = \frac{1}{J_T}(J_p + \frac{1}{4}m_pL_p^2)\left(\tau - B_r\dot{\theta}\right) + \frac{1}{2J_T}m_pL_pL_r(\frac{1}{2}m_pL_pg\alpha - B_p\dot{\alpha}). \tag{1-38}$$

Expanding the equation and collecting similar terms gives us:

$$\ddot{\theta} = \frac{1}{J_T}(-(J_p + \frac{1}{4}m_pL_p^2)B_r\dot{\theta} - \frac{1}{2}m_pL_pL_rB_p\dot{\alpha} + \frac{1}{4}m_p^2L_p^2L_r\,g\alpha + (J_p + \frac{1}{4}m_pL_p^2)\tau) \tag{1-39}$$

For the second equation, the multiplication of the matrix leads to:

$$\ddot{\alpha} = \frac{1}{2J_T}m_pL_pL_r\left(\tau - B_r\dot{\theta}\right) + \frac{1}{J_T}\left(J_r + m_pL_r^2\right)(\frac{1}{2}m_pL_pg\alpha - B_p\dot{\alpha}) \tag{1-40}$$

$$\ddot{\alpha} = \frac{1}{J_T}(-\frac{1}{2}m_pL_pL_rB_r\dot{\theta} - \left(J_r + m_pL_r^2\right)B_p\dot{\alpha} + \frac{1}{2}m_pL_pg(J_r + m_pL_r^2)\alpha + \frac{1}{2}m_pL_pL_r\tau) \tag{1-41}$$

To obtain the linear state space of the rotating inverted pendulum system. You have to find the matrix A and B, C and D.

From the state defined in the equation (IV.36), it is given that $\dot{x}_1 = x_3$ et $\dot{x}_2 = x_4$, We replace the state x by the equations of motion found.

Where (as shown in equation IV.36) we have $\theta = x_1$, $\alpha = x_2$, $\dot{\theta} = x_3$,$\dot{\alpha} = x_4$. the matrix A and B for $\dot{x} = Ax + Bu$ can then be found.

Substituting x in the equation and the equation gives:

$$\dot{x}_3 = \frac{1}{J_T}(-(J_p + \frac{1}{4}m_p L_p^2)B_r x_3 - \frac{1}{2}m_p L_p L_r B_p x_4 + \frac{1}{4}m_p^2 L_p^2 L_r g x_2 + (J_p + \frac{1}{4}m_p L_p^2)u) \tag{1-42}$$

And

$$\dot{x}_4 = \frac{1}{J_T}(-\frac{1}{2}m_p L_p L_r B_r x_3 - (J_r + m_p L_r^2)B_p x_4 + \frac{1}{2}m_p L_p g(J_r + m_p L_r^2)x_2 + \frac{1}{2}m_p L_p L_r u) \tag{1-43}$$

The matrix A et B in equation $\dot{x} = Ax + Bu$:

$$A = \frac{1}{J_T}\left[0\ 0\ J_T\ 0\ 0\ 0\ 0\ J_T\ 0\ \frac{1}{4}m_p^2 L_p^2 L_r\ g\ -(J_p + \frac{1}{4}m_p L_p^2)\ B_r\ \ -\frac{1}{2}m_p L_p L_r B_p\ 0\ \frac{1}{2}m_p L_p g(J_r + m_p L_r^2)\ -\frac{1}{2}m_p L_p L_r B_r\ -(J_r + m_p L_r^2)B_p\right] \tag{1-44}$$

$$B = \frac{1}{J_T}\left[0\ 0\ J_p + \frac{1}{4}m_p L_p^2\ \frac{1}{2}m_p L_p L_r\right]$$

1.9 Validation of QUBE-SERVO model using states:

The coefficient matrices of state space representation are calculated from equations derived in analytical modeling. Using the values of parameters from Table (table1-2), these matrices (1-45) are calculated by(table1-2),

(1-46)

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 143.2751 & -0.0109 & 0 \\ 0 & 258.6091 & -0.0107 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 48.7275 \\ 48.1493 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The output obtained from the state space model of QUBE-Servo RIP system is compared with angular positions obtained from the Simulation model .Square wave is used as test input for comparison. Simulink model of the state space validation along with the results are shown in. Figure 1-14 and Figure 1-15. After comparing, it can be seen that the model built using **Matlab Simulink** is similar to QUBE-Servo [1].

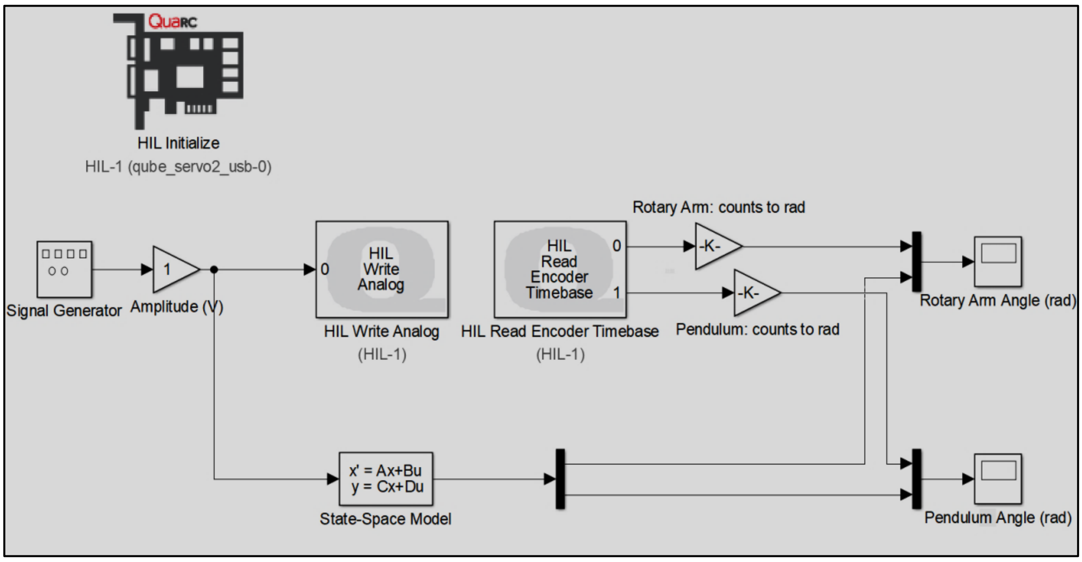


Figure 1-14 State space validation

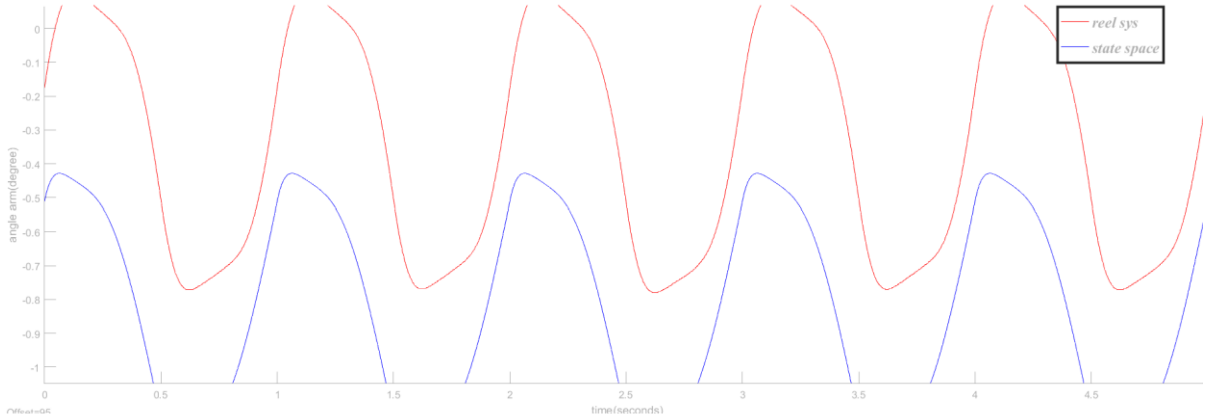


Figure 1-15 Comparison of angular displacement of rotary arm

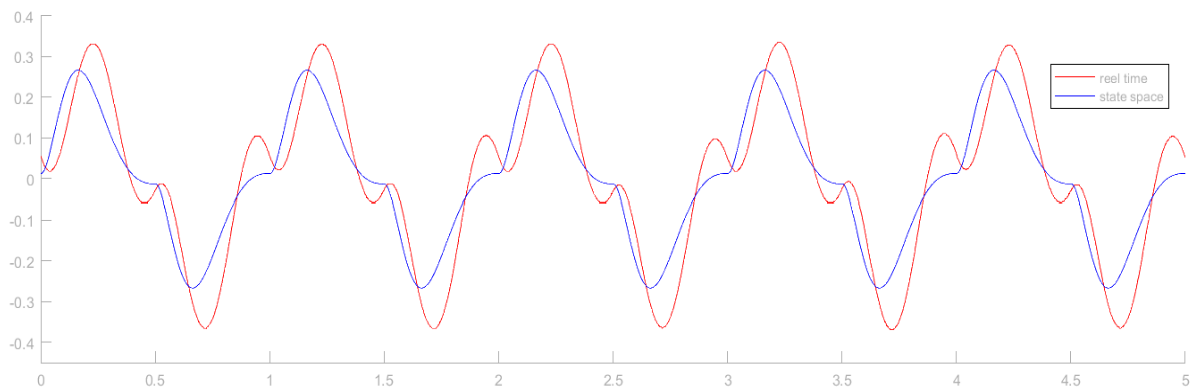


Figure 1-16 Comparison of angular displacement of pendulum

1.10 Conclusion

In this chapter, we conducted a detailed study about the rotary inverted pendulum and on the development of its mathematical form then we used the of Euler -Lagrange method to find and establish the dynamic equation. Finally, we have represented this system on state space by linearization of the model around the point of balance.

In the next chapter we will take a look at the application of the two commands the quadratic linear control and LQR genetic in order to find out which has the best impact on the control system.

CHAPTER II

Control System

Chapter 2: Control System

1.11 Introduction

The inverted cart-pendulum system is an under actuated mechanical system. It has two degrees of freedom and one control input the inverted pendulum is nonlinear system with only one control input which is voltage applied to DC motor and two outputs one is angular rotation of pendulum arm and the rotary pendulum inverted angle. The inverted pendulum is considered as a platform to study real world non-linear control problems; we have different control design techniques are being developed to balance the inverted pendulum in an upright position while the arm moving [27]. On this chapter we will talk about optimal control used to control the pendulum We will take an overview of optimal control and the basics and fundamentals of linear quadratic regulator genetic algorithms which we use it to optimize the LQR controller.

1.12 Classical and Modern Control

The classical (conventional) control theory concerned with single input and single output is mainly based on Laplace transforms theory and its use in system representation in block diagram form, we see that

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{2-1}$$

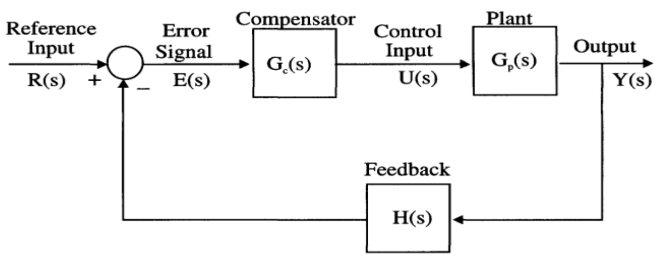


Figure 2-1 Classical Control Configuration [4].

where s is Laplace variable and we used (2-1)

$$G(s) = Gc(s)Gp(s) \tag{2-2}$$

Note that

1. the input u(t) to the plant is determined by the error e(t) and the compensator, and
2. all the variables are not readily available for feedback. In most cases only one output variable is available for feedback. The modern control theory concerned with multiple inputs and multiple outputs (MIMO) is based on state variable representation in terms of a set of first order differential (or difference) equations. Here, the system (plant) is characterized by state variables, say, in linear, time invariant form as

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t) \end{aligned} \tag{2-3}$$

where, dot denotes differentiation with respect to (w.r.t.) t, x(t), u(t), and y(t) are n, r, and m dimensional state, control, and output vectors respectively, and A is nxn state, B is nxr input, C is mxn output, and D is mxr transfer matrices. Similarly, a nonlinear system is characterized by

$$\begin{aligned} x\dot{(t)} &= f(x(t), u(t), t \\ y(t) &= g(x(t), u(t), t \end{aligned} \tag{2-4}$$

The modern theory dictates that all the state variables should be fed back after suitable weighting. We see from Figure (2-2) that in modern control configuration,

1. the input u(t) is determined by the controller (consisting of error detector and compensator) driven by system states x(t) and reference signal r (t),
2. all or most of the state variables are available for control, and
3. it depends on well-established matrix theory, which is amenable for large scale computer simulation [4].

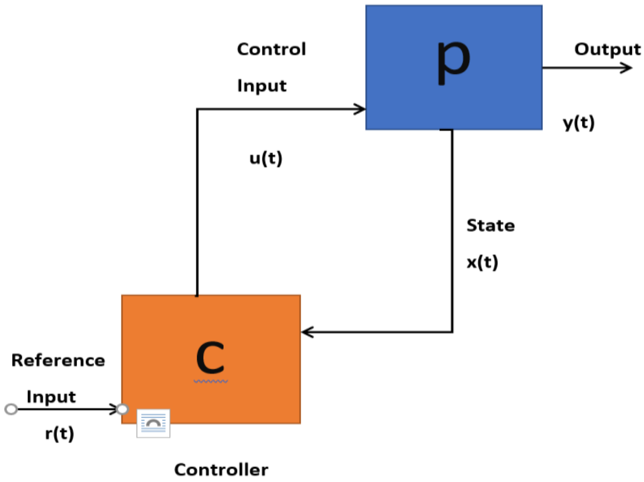


Figure 2-2 Modern Control Configuration

The fact that the state variable representation uniquely specifies the transfer function while there are a number of state variable representations for a given transfer function, reveals the fact that state variable representation is a more complete description of a system. Figure 2-2 shows components of a modern control system. It shows three components of modern control and their important contributors. The first stage of any control system theory is to obtain or formulate the dynamics or modelling in terms of dynamical equations such as differential or difference equations. The system dynamics is largely based on the Lagrange function. Next, the system is analyzed for its performance to mainly find out stability of the system and the contributions of Lyapunov to stability theory are well known. Finally, if the system performance is not according to our specifications, we resort to design. In optimal control theory, the design is usually with respect to a performance index. We notice that although the concepts such as Lagrange function and V function of Lyapunov are old, the techniques using those concepts are modern. Again, as the phrase modern usually refers to time and what is modern today becomes ancient after a few years, a more appropriate thing is to label them as optimal control, nonlinear control, adaptive control, robust control and so on [4].

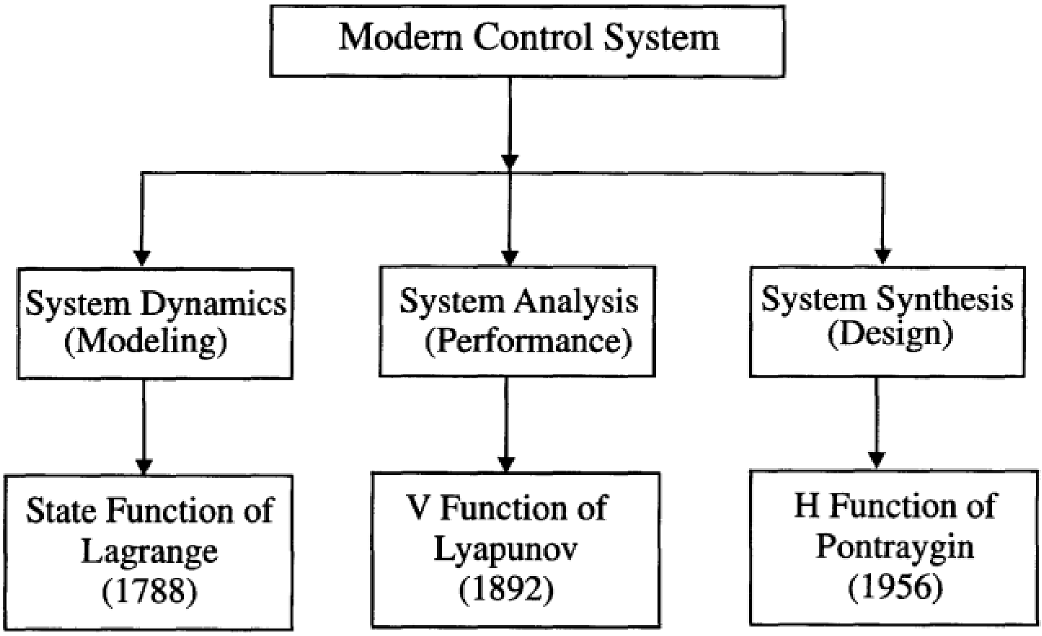


Figure 2-3 Components of a Modern Control System [4].

1.13 Optimization

Optimization is a very desirable feature in day-to-day life. We like to work and use our time in an optimum manner, use resources optimally and so on. The subject of optimization is quite general in the sense that it can be viewed in different ways depending on the approach (algebraic or geometric), the interest (single or multiple), the nature of the signals (deterministic or stochastic), and the stage (single or multiple) used in optimization. This is shown in Figure 2-4. As we notice that the calculus of variations is one small area of the big picture of the optimization field, and it forms the basis for our study of optimal control systems. Further, optimization can be classified as static optimization and dynamic optimization [4].

1.13.1 Static Optimization

It is concerned with controlling a plant under steady state conditions, i.e., the system variables are not changing with respect to time. The plant is then described by algebraic equations. Techniques used are ordinary calculus, Lagrange multipliers, linear and nonlinear programming.

1.13.2 Dynamic Optimization

It concerns with the optimal control of plants under dynamic conditions, i.e., the system variables are changing with respect to time and thus the time is involved in system description. Then the plant is described by differential

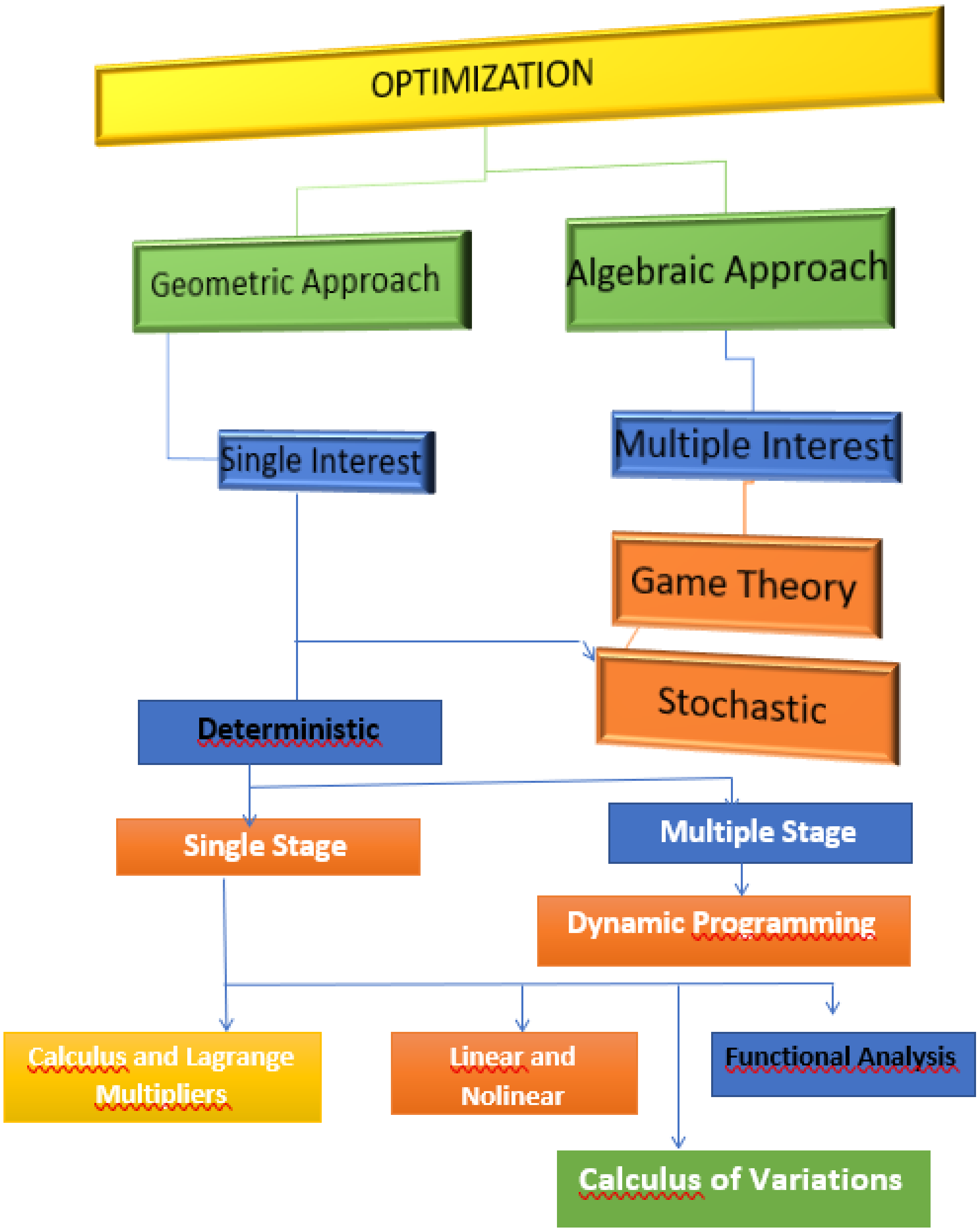


Figure 2-4 Overview of Optimization

(Or difference) equations. Techniques used are search techniques, dynamic programming, variational calculus (or calculus of variations) and Pontryagin principle.

1.14 Optimal Control

The main objective of optimal control is to determine control signals that will trigger a process (plant) to satisfy some physical constraints and at the same time extremist (maximize or minimize) a chosen performance criterion (performance index or cost function). Referring to Figure 2- , we are interested in finding the optimal control u*(t) (* indicates optimal condition) that will drive the plant P from initial state to final state with some constraints on controls and states and at the same time eternizing the given performance index J. The formulation of optimal control problem requires

1. a mathematical description (or model) of the process to be controlled (generally in state variable form),
2. a specification of the performance index, and
3. a statement of boundary conditions and the physical constraints on the states and/or controls [4].

1.14.1 Plant

For the purpose of optimization, we describe a physical plant by a set of linear or nonlinear differential or difference equations. For example, a linear time-invariant system is described by the state and output relations (2-3) and (2-4) and a nonlinear system by (2-5) and (2-6) [4].

1.14.2 Performance Index

Classical control design techniques have been successfully applied to linear, time-invariant, single-input, single output (8180) systems. Typical performance criteria are system time response to step or ramp up input characterized by rise time, settling time, peak overshoot, and steady state accuracy; and the frequency response of the system characterized by gain and phase margins, and bandwidth. In modern control theory, the optimal control problem is to find a control which causes the dynamical system to reach a target or follow a state variable (or trajectory) and at the same time extremist a performance index which may take several forms as described below[4].

1.14.2.1 Performance Index for Time-Optimal Control System:

We try to transfer a system from an arbitrary initial state x(t0) to a specified final state x(tf) in minimum time. The corresponding performance index (PI) is

$$J = \int_{t_0}^{t_e} \quad dt = t_f - t_0 = t^*. \tag{2-5}$$

1.14.2.2 Performance Index for Fuel-Optimal Control System:

Consider a spacecraft problem. Let u(t) be the thrust of a rocket engine and assume that the magnitude I u(t) I of the thrust is proportional to the rate of fuel consumption. In order to minimize the total expenditure of fuel, we may formulate the performance index as

$$J = \int_{t_0}^{t_e} \quad |U(t)|dt \tag{2-6}$$

1.14.2.3 Performance Index for Minimum-Energy Control System:

Consider Ui (t) as the current in the ith loop of an electric network. Then $\sum_{i=1}^{m} \quad Ui^2(t)$ri (where, ri is the resistance of the ith loop) is the total power or the total rate of energy expenditure of the network. Then, for minimization of the total expended energy, we have a performance criterion as

$$J = \int_{t_0}^{t_e} \sum_{i=1}^{m} \quad Ui^2(t)dt \tag{2-7}$$

1.14.2.4 Performance Index for General Optimal Control System:

Combining the above formulations, we have a performance index in general form as

$$J \ = \ X'(tf)Fx(tf) + \int_{t_0}^{t_e} \quad [X'(t)QX(t) \ + \ u'(t)Ru(t)]dt \tag{2-8}$$

$$J \ = \ S(x(tf), tf) + \int_{t_0}^{t_e} \quad V(x(t), u(t), t)dt \tag{2-9}$$

where, R is a positive definite matrix, and Q and F are positive semidefinite matrices, respectively. Note that the matrices Q and R may be time varying. The particular form of performance index (2-8) is called quadratic (in terms of the states and controls) form.

The problems arising in optimal control are classified based on the structure of the performance index J. If the PI (2-9) contains the terminal cost function S(x(t), u(t), t) only, it is called the Mayer problem, if the PI (2-9) has only the integral cost term, it is called the Lagrange problem, and the problem is of the Bolza type if the PI contains both the terminal cost term and the integral cost term as in (2-9). There are many other forms of cost functions depending on our performance specifications. However, the above-mentioned performance indices (with quadratic forms) lead to some very elegant results in optimal control systems.

1.14.3 Constraints

The control u(t) and state x(t) vectors are either unconstrained or constrained depending upon the physical situation. The unconstrained problem is less involved and gives rise to some elegant results. From the physical considerations, often we have the controls and states, such as currents and voltages in an electrical circuit, speed of a motor, thrust of a rocket, constrained as

X+≤x(t)≤X-, and U+≤ u(t) ≤U-

where, +, and - indicate the maximum and minimum values the variables can attain.

1.14.4 Formal Statement of Optimal Control System

Let us now formally state the optimal control problem. The optimal control problem is to find the optimal control u*(t) (* indicates extremal or optimal value) which causes the linear time-invariant plant (system)

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2-10}$$

to give the trajectory x* (t) that optimizes or extremizes (minimizes or maximizes) a performance index

$$J = X'(tf)Fx(tf) + \int_{t_0}^{t_e} [X'(t)QX(t) + u'(t)Ru(t)]dt \tag{2-11}$$

or which causes the nonlinear system

$$\dot{x}(t) = f(x(t), u(t), t) \tag{2-12}$$

to give the state x*(t) that optimizes the general performance index

$$J = S(x(tf), tf) + \int_{t_0}^{t_e} V(x(t), u(t), t)dt \tag{2-13}$$

with some constraints on the control variables u(t) and/or the state variables x(t) given by (2-10). The final time tf may be fixed, or free, and the final (target) state may be fully or partially fixed or free. The entire problem statement is also shown pictorially in Figure 2-5. Thus we are basically interested in finding the control u*(t) which when applied to the plant described by (2-11) or (2-13), gives an optimal performance index J* described by (2-12) or (2-14). The optimal control systems are studied in three stages.

1. In the first stage, we just consider the performance index of the form (2-14) and use the well-known theory of calculus of variations to obtain optimal functions.
2. In the second stage, we bring in the plant (2-11) and try to address the problem of finding optimal control u*(t) which will drive the plant and at the same time optimize the performance index (2-12). Next, the above topics are presented in discrete time domain.
3. Finally, the topic of constraints on the controls and states (2-10) is considered along with the plant and performance index to obtain optimal control[4].

1.15 Linear quadratic regulator:

Linear quadratic control: LQ or LQR for "linear quadratic regulator", is an optimal control law u(t) in closed loop which allows to ensure the desired performance. It is one of the most popular design methods widely answered for system control and stability according to different criteria. when the system is linear and the criterion to be minimized is quadratic, this is the linear quadratic controller, this control is defined as being an optimal controller by state feedback [3]. The principle of the LQR control is presented in the figure

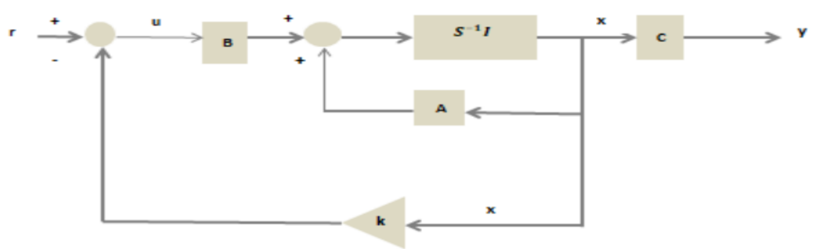


Figure 2-5 Principle of an LQR controller [2].

1.15.1 Properties and Use of the LQR

- **Static Gain:** LQR generates a gain matrix static *K*, which is not a dynamical system. Hence, the order of the closed-loop system is the same as that of the plan.
- **Output Variables:** When we want to conduct output regulation (and not state regulation), we set $Q = C^T Q' C$.
- **Robustness:** LQR achieves infinite gain margin [12].

1.15.2 Concept of linear quadratic regulator

LQR controller is an optimal state feedback controller that is concerned with operating dynamic system at minimum cost and time. It provides optimal feedback gains to enable close loop system stability and high performance, note that there are two points of balance for inverted pendulums, 180 degrees (stable) and 0 degrees (unstable). A schematic diagram of the system with linearization and a K regulator is shown in figure 2-6

LQR controller has designed using algebraic Riccati equation given by Eq

$$A^T P + P A + PBR^{-1}B^T P + Q = 0 \quad (2\text{-}14)$$

the Q and R,are state control matrices, A is the state matrix, B is the input matrix, and P is the transformation matrix. State feedback gain matrix is indicated by

$$K = R^{-1}B^T P \quad (2\text{-}15)$$

the performance index J is found and controller has to be designed to minimize

$$J = \int (X^T QX + U^T RU)dt \quad (2\text{-}16)$$

According to this boundary condition and by backward integration of Riccati Equation, optimal feedback gain can be calculated online

$$U = -KX \quad (2\text{-}17)$$

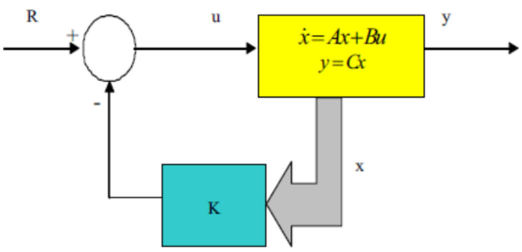


Figure 2-6 Block diagram for the realization of LQR [29].

This form is that of state feedback. The MATLAB command LQR returns a set of gains calculated on the basis of the matrices A and B and the design matrices Q and R. Where Q is State weighted matrix and R is control weighted matrix, that penalize certain states or control inputs. In the design, the weighting parameters of the optimal state feedback controller are chosen as follows:

$$\mathbf{Q}=\begin{bmatrix} \boldsymbol{Q1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \boldsymbol{Q2} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \boldsymbol{Q3} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \boldsymbol{Q4} \end{bmatrix} \boldsymbol{R} = \boldsymbol{Q0} \tag{2-18}$$

Then we vary the matrix diagonal Q in an interval that we will define and hold R to 1, we can then choose which state variable the command places more emphasis on by increasing the associated qi parameter[3].

1.15.3 Choice of weighting matrices

Linear Quadratic Regulator (LQR) is an optimal control method. The main objective of optimal control is to determine control signals that will cause a Process (Plant) to satisfy some physical constraints and at the same time extremize (maximize or minimize) a chosen performance criterion (performance index or cost function). The synthesis of the optimal controller gain matrices is based directly on the Q and R weighting matrices. The compositions of Q and R elements have great influences on system performance. The designer is free to select the matrices Q and R, but the selection of matrices Q and R is normally based on an iterative procedure using experience and physical understanding of the problems involved. However, to simplify the determination of these matrices, it is generally preferred to render invariants and diagonals. Indeed, by proceeding thus, we are reduced to the choice of m scalars for R and n scalars for Q. There are two methods for choosing them:

➜ The first method is Bryson's rule which suggests choosing diagonal weighting matrices, whose diagonal coefficients are equal to the square of the inverse of the desired maximum deviation on the corresponding variable. Bryson also indicates that this rule only provides initial values, which can then be imp

roved by successive simulations.

In the second method the weighting matrices Q and R can be chosen symmetrical, diagonal. At the start, we generally choose weightings equal to the identity matrices which we can then improve by successive simulations until a satisfactory corrector is obtained [13].

The choice of weighting matrix favors energy saving [29].

1.15.4 feedback gain matrix

the optimal control feedback gain matrix can be obtained by Riccati equation. Its form is as follows [14]:

$$K = R^{-1}(B^T P + N^T) \tag{2-19}$$

P is obtained by solving the Riccati equation:

$$K = A^T(P - P\, B(R + B^T PB)^{-1} B^T P)A + Q = 0 \tag{2-20}$$

The differential equation describing the behavior of the closed-loop system:

$$\dot{x}\,(k) = (A - BK)\, x(K) \tag{2-21}$$

1.15.5 LQR control law calculation:

An explicit diagram of the optimal control system with criterion

quadratic is given as follows:

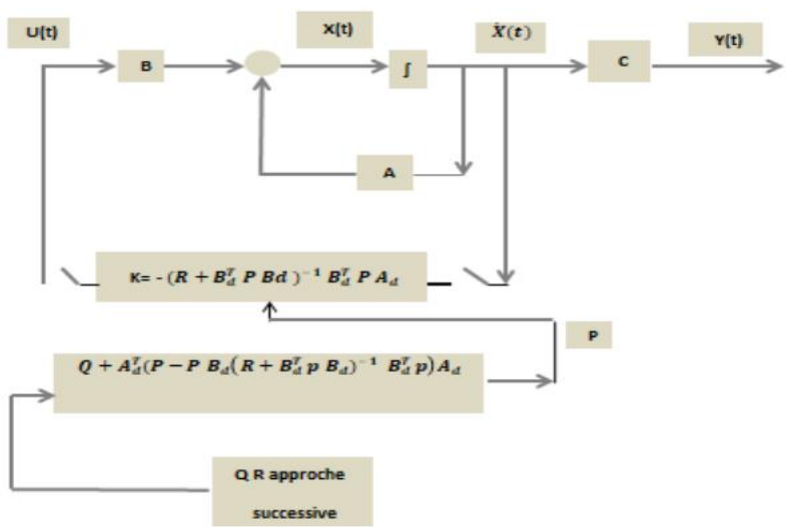


Figure 2-7 Optimal closed-loop control [2].

The difference with the control by placement of poles appears in the fact that the gain matrix K is calculated according to the constraints imposed on the system, constraints that are expressed through the R and Q weightings.

1.16 Genetic Algorithm

Optimization techniques are the techniques used to discover the best solution out of all the possible solutions available under the constraints present, Optimization refers to finding the values of inputs in such a way that we get the “best” output values. The definition of “best” varies from problem to problem, but in mathematical terms, it refers to maximizing or minimizing one or more objective functions, by varying the input parameters.

The set of all possible solutions or values which the inputs can take make up the search space. In this search space, lies a point or a set of points which gives the optimal solution. The aim of optimization is to find that point or set of points in the search space. The genetic algorithm is one such optimization algorithm built based on the natural evolutionary process of our nature. The idea of Natural Selection and Genetic Inheritance is used here. Unlike other algorithms, it uses guided random search, i.e., finding the optimal solution by starting with a random initial cost function and then searching only in the space with the least cost (in the guided direction). Suitable when you are working with huge and complex datasets, we have a pool or a population of possible solutions to the given problem. These solutions then undergo recombination and mutation (like in natural genetics), producing new children, and the process is repeated over various generations. Each individual (or candidate solution) is assigned a fitness value (based on its objective function value) and the fitter individuals are given a higher chance to mate and yield more “fitter” individuals. This is in line with the Darwinian Theory of “Survival of the Fitness”. In this way we keep “evolving” better individuals or solutions over generations, till we reach a stopping criterion [15].

Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well.

the important criteria for GA approach can be formulated as given below:

- Completeness: Any solution should have its encoding

- Soundness: Any code (produced by genetic operators) should have its corresponding solution

- Non redundancy: Codes and solutions should correspond one to one

- Characteristic perseverance: Offspring should inherit useful characteristics from parents [16].

1.16.1 Advantages and the limitation of Genetic Algorithms:

Genetic Algorithms have various advantages which have made them immensely popular. These include

- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Always gets an answer to the problem, which gets better over the time.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Useful when the search space is very large and there are a large number of parameters involved.
- Has very good parallel capabilities [17].

The limitation of genetic algorithm includes,

1. The problem of identifying fitness function
2. Definition of representation for the problem
3. Premature convergence occurs
4. Cannot easily incorporate problem specific information
5. Cannot use gradients.
6. The problem of choosing the various parameters like the size of the population, mutation rate, cross over rate, the selection method and its strength.
7. Not good at identifying local optimal [17].

1.16.2 Concept of Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome[18].

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosomes of the population. Following is the foundation of GAs based on this analogy

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from “fittest” parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
4. Thus, each successive generation is more suited for their environment[19].

1.16.3 Genotype Representation

One of the most important decisions to make while implementing a genetic algorithm is deciding the representation that we will use to represent our solutions. It has been observed that improper representation can lead to poor performance of the GA.

Therefore, choosing a proper representation, having a proper definition of the mappings between the phenotype and genotype spaces is essential for the success of a GA.

In this section, we present some of the most commonly used representations for genetic algorithms. However, representation is highly problem specific and the reader might find that another representation or a mix of the representations mentioned here might suit his/her problem better [20].

1.16.3.1 Binary Representation

This is one of the simplest and most widely used representation in GAs. In this type of representation, the genotype consists of bit strings.

For some problems when the solution space consists of Boolean decision variables – yes or no, the binary representation is natural. Take for example the 0/1 Knapsack Problem. If there are n items, we can represent a solution by a binary string of n elements, where the x^{th} element tells whether the item x is picked (1) or not (0).

0	0	1	0	1	1	1	0	0	1

For other problems, specifically those dealing with numbers, we can represent the numbers with their binary representation. The problem with this kind of encoding is that different bits have different significance and therefore mutation and crossover operators can have undesired consequences. This can be resolved to some extent by using **Gray Coding,** as a change in one bit does not have a massive effect on the solution[21].

1.16.3.2 Real Valued Representation

For problems where we want to define the genes using continuous rather than discrete variables, the real valued representation is the most natural. The precision of these real valued or floating-point numbers is however limited to the computer.

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9

1.16.3.3 Integer Representation

For discrete valued genes, we cannot always limit the solution space to binary ‘yes’ or ‘no’. For example, if we want to encode the four distances – North, South, East and West, we can encode them as **{0,1,2,3}**. In such cases, integer representation is desirable.

1	2	3	4	3	2	4	1	2	1

1.16.3.4 Permutation Representation

In many problems, the solution is represented by an order of elements. In such cases permutation representation is the most suited.

A classic example of this representation is the travelling salesman problem (TSP). In this the salesman has to take a tour of all the cities, visiting each city exactly once and come back to the starting city. The total distance of the tour has to be minimized. The solution to this TSP is naturally an ordering or permutation of all the cities and therefore using a permutation representation makes sense for this problem[21].

1	5	9	8	7	4	2	3	6	0

1.16.4 Parents

Everything starts with the parents. Two of them to be more exactly. They provide the necessary structure and information to create the new children, as we said in the beginning, we used a Deep Learning Neural Network to explain Generic Algorithm and now is time to add it into the explanation. For education purpose, we create a Full Dense Neural Network with 5 Inputs and 3 Outputs, doesn’t matter the purpose of this Neural Network.

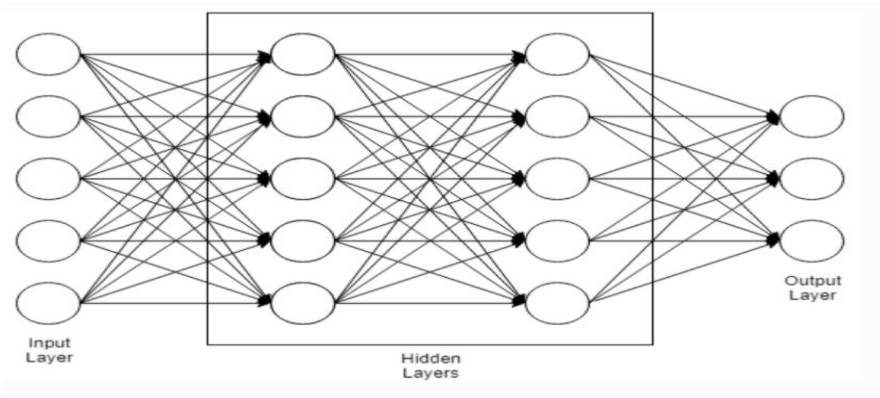


Figure 2-8 **Neural Network**

Initially our network has 2 fixed layers (input and output) and a variable number of hidden layers each with a variable number of neurons and with unknown Activation Function. We can call these combinations as Chromosome invoking the real name of that “” *object/thing/element* “” that’s carry our biological information.

1.16.5 Fitness Function

The fitness function simply defined is a function which takes a candidate solution to the problem as input and produces as output how “fit” our how “good” the solution is with respect to the problem in consideration.

Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a GA and make it exceptionally slow.

In most cases the fitness function and the objective function are the same as the objective is to either maximize or minimize the given objective function. However, for more complex problems with multiple objectives and constraints, an Algorithm Designer might choose to have a different fitness function.

- A fitness function should possess the following characteristics
- The fitness function should be sufficiently fast to compute.

It must quantitatively measure how fit a given solution is or how fit individuals can be produced from the given solution.

In some cases, calculating the fitness function directly might not be possible due to the inherent complexities of the problem at hand. In such cases, we do fitness approximation to suit our needs [15].

1.16.6 Mechanism of a Genetic Algorithm

As you can see in the diagram, the Genetic Algorithm implementation consist on 8 distinct steps.

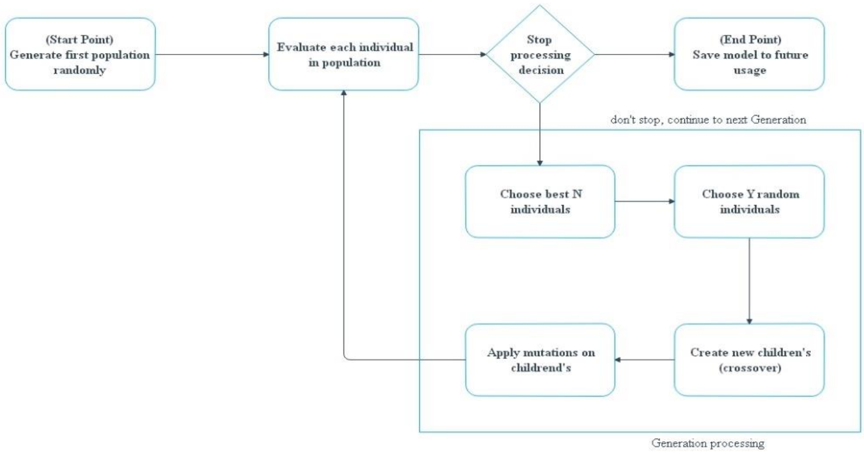


Figure 2-9 Diagram of Generic Genetic Algorithm Implementation

the basic four steps used in simple Genetic Algorithm to solve a problem are,

1. The representation of the problem
2. The fitness calculation
3. Various variables and parameters involved in controlling the algorithm
4. The representation of result and the way of terminating the algorithm [23].

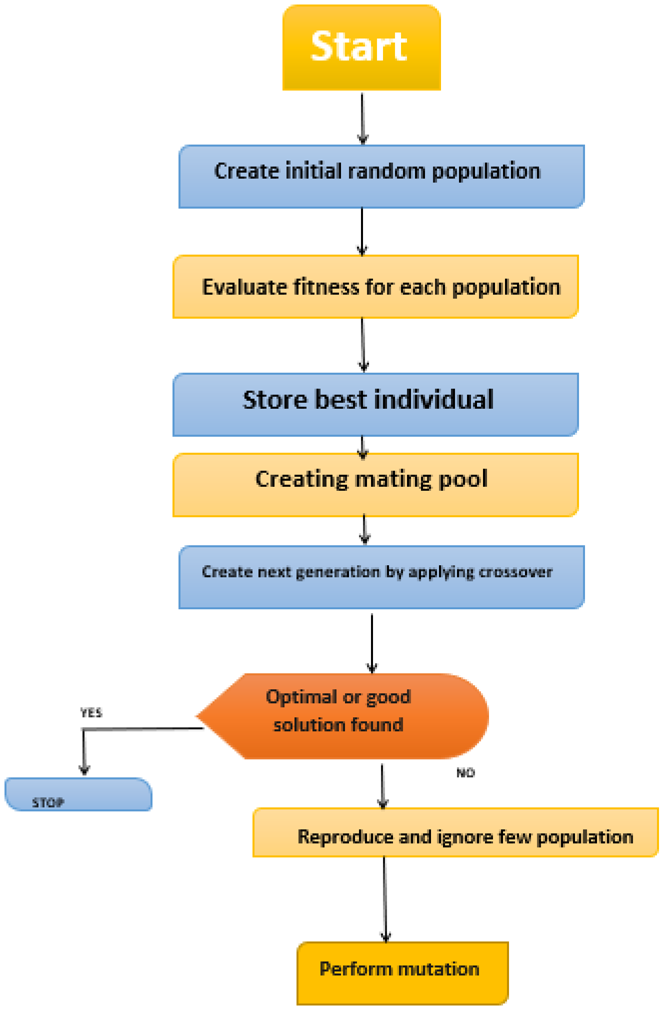


Figure 2-10 Flowchart of genetic algorithm

1.16.7 Structure of genetic algorithm

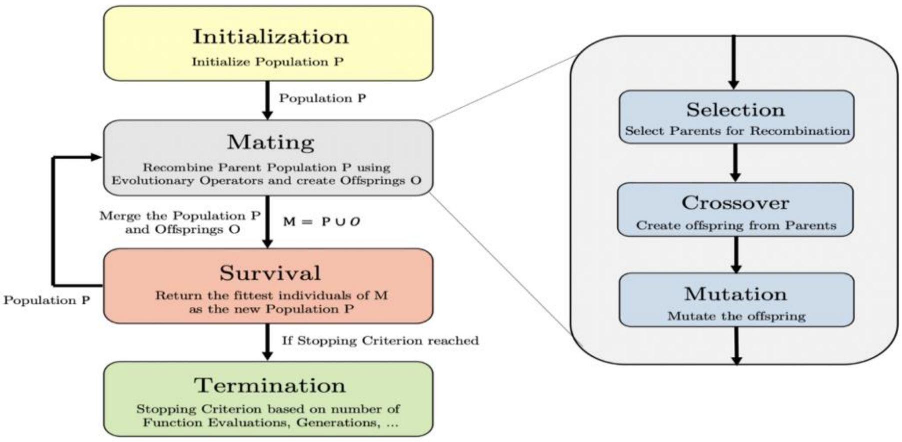


Figure 2-11 the Genetic Algorithm phases [24].

the different phases of the Genetic Algorithm represented in:

1.16.7.1 Initialization of Population (Coding)

- Every gene represents a parameter (variables) in the solution. This collection of parameters that forms the solution is the chromosome. Therefore, the population is a collection of chromosomes.
- Order of genes on the chromosome matters.
- Chromosomes are often depicted in binary as 0’s and 1’s, but other encodings are also possible.

1.16.7.2 2. Fitness Function

- We have to select the best ones to reproduce offspring out of the available chromosomes, so each chromosome is given a fitness value.
- The fitness score helps to select the individuals who will be used for reproduction.

1.16.7.3 Selection

- This phase’s main goal is to find the region where getting the best solution is more.
- Inspiration for this is from the survival of the fittest.
- It should be a balance between exploration and exploitation of search space.
- GA tries to move the genotype to higher fitness in the search space.
- Too strong fitness selection bias can lead to sub-optimal solutions.

- Too little fitness bias selection results in an unfocused search.
- Thus, Fitness proportionate selection is used, also known as roulette wheel selection, as a genetic operator used in genetic algorithms to select potentially useful recombination solutions.

1.16.7.4 Reproduction

Generation of offspring happen in 2 ways:

a) Crossover

The crossover operator is analogous to reproduction and biological crossover. In this more than one parent is selected and one or more off-springs are produced using the genetic material of the parents. Crossover is usually applied in a GA with a high probability

There are 3 major types of crossovers.

- **Single Point Crossover:** A point on both parents' chromosomes is picked randomly and designated a 'crossover point'. Bits to the right of that point are exchanged between the two parent chromosomes.
- **Two-Point Crossover:** Two crossover points are picked randomly from the parent chromosomes. The bits in between the two points are swapped between the parent organisms.
- **Uniform Crossover:** In a uniform crossover, typically, each bit is chosen from either parent with equal probability.

The new offspring are added to the population.

b) Mutation

In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – $\boldsymbol{p_m}$. If the probability is very high, the GA gets reduced to a random search. Mutation is the part of the GA which is related to the "exploration" of the search space. It has been observed that mutation is essential to the convergence of the GA while crossover is not.

we describe some of the most commonly used mutation operators.

- **Bit Flip Mutation:** In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

- **Random Resetting**

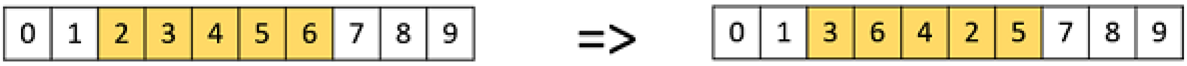
Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

- **Swap Mutation**

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation-based encodings.

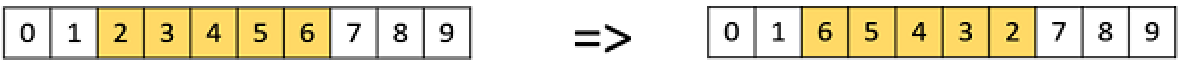
- **Scramble Mutation**

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.



- **Inversion Mutation**

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.



1.16.7.5 Convergence (when to stop)

Few rules which are followed which tell when to stop is as follows:

When there is no improvement in the solution quality after completing a certain number of generations set beforehand.

When a hard and fast range of generations and time is reached.

Till an acceptable solution is obtained [24].

1.16.8 Application of Genetic Algorithm

Genetic Algorithms are primarily used in optimization problems of various kinds, but they are frequently used in other application areas as well. We list some of them

- **Multimodal Optimization** − GAs are obviously very good approaches for multimodal optimization in which we have to find multiple optimum solutions.
- **Neural Networks** − GAs are also used to train neural networks, particularly recurrent neural networks.
- **Optimization** − Genetic Algorithms are most commonly used in optimization problems wherein we have to maximize or minimize a given objective function value under a given set of constraints. The approach to solve Optimization problems has been highlighted throughout the tutorial.
- **Economics** − GAs are also used to characterize various economic models like the cobweb model, game theory equilibrium resolution, asset pricing, etc.
- **Parallelization** − GAs also have very good parallel capabilities, and prove to be very effective means in solving certain problems, and also provide a good area for research.
- **Image Processing** − GAs are used for various digital image processing (DIP) tasks as well like dense pixel matching.
- **Vehicle routing problems** − With multiple soft time windows, multiple depots and a heterogeneous fleet.
- **Scheduling applications** − GAs are used to solve various scheduling problems as well, particularly the time tabling problem.
- **Machine Learning** − as already discussed, genetics-based machine learning (GBML) is a niche area in machine learning.
- **Robot Trajectory Generation** − GAs have been used to plan the path which a robot arm takes by moving from one point to another.
- **Parametric Design of Aircraft** − GAs have been used to design aircrafts by varying the parameters and evolving better solutions.
- **DNA Analysis** − GAs have been used to determine the structure of DNA using spectrometric data about the sample.
- **Traveling salesman problem and its applications** − GAs have been used to solve the TSP, which is a well-known combinatorial problem using novel crossover and packing strategies.

1.17 Optimized LQR Controller Using Genetic Algorithm

In LQR problem, the weighting matrices Q and R have significant effect on the performance of the controller. Finding the best values for Q and R are highly time consuming by implementing computer simulations or trial and error methods [26]. For effective solution some intelligent optimization techniques can be applied. The main objective of the genetic algorithm implemented here is to determine the weighting matrices Q and R in order to have better performance. Q and R are usually represented as diagonal matrices[22].

$$Q = \begin{bmatrix} \boldsymbol{Q1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} \\ \boldsymbol{1} & \boldsymbol{Q2} & \boldsymbol{1} & \boldsymbol{1} \\ \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{Q3} & \boldsymbol{1} \\ \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{Q4} \end{bmatrix}; R = Q0 \tag{2-22}$$

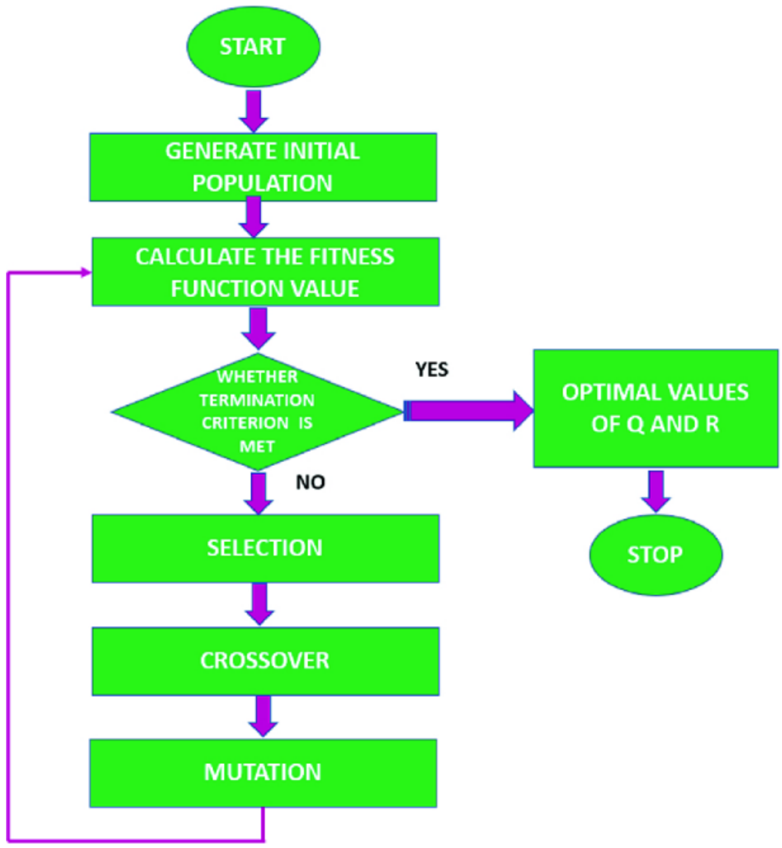


Figure 2-12 Flowchart for optimizing LQR using GAs [25].

1.18 Conclusion

In this chapter, we presented the principle of LQR control and genetic algorithms. the LQR controller is efficient in stabilizing the system thus the adjustment of the control is rather simple, since the two parameters of weighting are the only ones to be defined by the user to obtain a compromise between desired performance and allowable control effort.

we use genetic algorithms to optimize the controller, and to get the state weighing matrix and control weighing matrix in the light of the time domain index, and to solve Riccati equation to get the optimal results.

Chapter III

Simulation and Results

Chapter 3: Simulation and Results

1.19 Introduction

In this chapter, we will apply different commands to our pendulum system, and we'll present our numerical result and graphical simulation using: **MATLAB/SIMULINK®.**

The results obtained are studied by applying different control strategies and are implemented and compared to the system that will stabilize it around its point of unstable equilibrium. To design such a control, it is necessary to respect the energy constraints, and in the face of all uncertainties presented in the models and all disturbances, we must ensure a certain robustness

The angular displacement of the rotary arm represented by and which represent the angular displacement pendulum from their reference point respectively are observed and shown by graphs for each control

we will start with the LQR control and to obtain more adequate results we will optimize the LQR using the genetic algorithm to study and integrate the influence of optimization algorithms.

1.20 Design of LQR controller for rotary inverted pendulum

To apply the LQR command, we ran a script file on **MATLAB**, The LQR method is a powerful and most used methods for control complex systems; the LQR algorithm calculates a control law U to find the optimal controller that minimizes a given cost function J, the design matrices Q and R contain the penalties on the deviations of the state variables from their set point and the actions of control, respectively. Q is a non-negative definite matrix that penalizes the departure of system states from the equilibrium, and R is a positive definite matrix that penalizes the control input.

The important point in this method is calculated gain matrix K the optimal feedback parameters of K matrix are taken by the cost function J.

$$J = \int (X^T\, QX + U^T\, RU)d \tag{3-1}$$

$$U = -KX \tag{3-2}$$

We defined the two matrix Q and R and calculated the gain matrix finely then we will implement LQR controller to stabilize the pendulum in the vertical position even in the presence of disturbance.

In our case the state vector x is defined:

$$x = [\theta_0, \alpha_0, \dot{\theta}_0, \dot{\alpha}_0] \tag{3-3}$$

The coefficient matrices of state space representation

$$A = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{143.2751} & \mathbf{-0.0109} & \mathbf{0} \\ \mathbf{0} & \mathbf{258.6091} & \mathbf{-0.0107} & \mathbf{0} \end{bmatrix} \tag{3-4}$$

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{48.7275} \\ \mathbf{48.1493} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

We take in first time Q and R as follow

$$Q = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, R = 1 \tag{3-5}$$

We changed the diagonal elements of Q many times with taking R=1, based on previous experiences to obtain the desired performance with several attempts. We deduce that the values of matrix taken from the article [30] below it gives best results and satisfying performance; Q and R is given as

$$Q = \begin{bmatrix} \mathbf{5} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{20} \end{bmatrix}, R = 1 \tag{3-6}$$

Using the system model from (3-4) and above weighting matrices, the state feedback controller gains obtained for robust LQR controller is:

$$K = [-2.2361 \quad 53.8606 \quad -1.6975 \quad 6.5259] \tag{3-7}$$

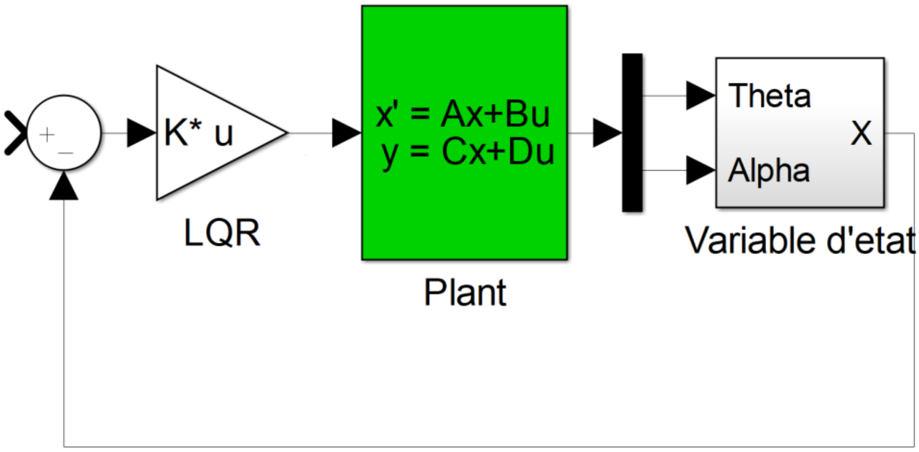


Figure 3-1 The LQR controller

After calculated gain matrix K, and we implemented the controller on simulation model of **QUB-Servo** developed in Simulink we illustrate three graphs the arm angle, pendulum angle, the command graph.

1.21 Genetic algorithm optimization

The selection of weighting matrix in design of the linear quadratic optimal controller is an important topic to design LQR controller, to select the weighting matrix for the optimal controller we will use genetic algorithm to optimize our control for best results. Genetic algorithm is adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. In this algorithm, the fitness function is used to evaluate individuals and reproductive success varies with fitness. In the design of the linear quadratic optimal controller, the fitness function has a relation to the anticipated step response of the system. Not only can the controller designed by this approach meet the demand of the performance indexes of linear quadratic controller, but also satisfy the anticipated step response of close-loop system.

The main objective of the genetic algorithm implemented here is to determine the weighting matrices Q and R in order to have better performance.

Applying the GAs, we obtained the fitness function that we use to determine the values of the weighting matrices Q and R to get gain matrix K.

After execution of the script in MATLAB, we obtained the fitness function as fellow

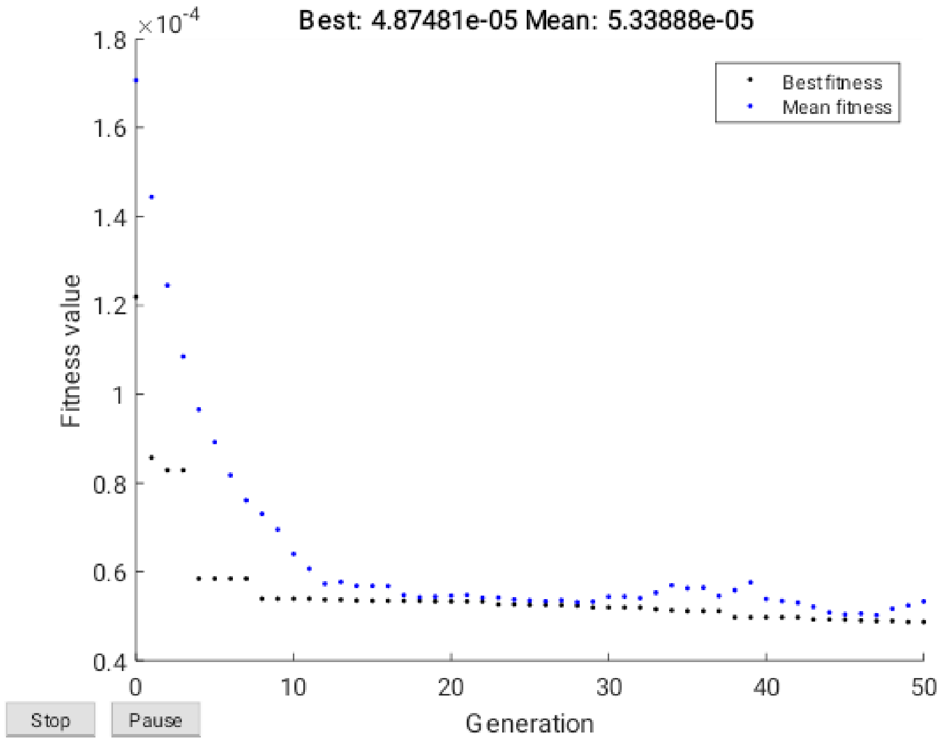


Figure 3-2 Fitness function

X=

0.1973 20.5213 8.6780 9.2072 0.5745

J=

0.000292752

The matrices Q and R of LQR optimized by genetic algorithms are:

$$Q = \begin{bmatrix} \mathbf{20.5213} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{8.6780} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{9.2072} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0.5745} \end{bmatrix}, \boldsymbol{R} = \mathbf{0.1973} \qquad (3\text{-}8)$$

Using the system model from (3-4) and above weighting matrices, the state feedback controller gains obtained for optimized LQR controller is:

$$\boldsymbol{K} = [\mathbf{-14.7979} \;\; \mathbf{178.4434} \;\; \mathbf{-6.8001} \;\; \mathbf{22.3600}] \qquad \mathbf{(3\text{-}9)}$$

After calculating the gain matrix K, and implementing the controller in the simulation model of **QUB-Servo** developed in Simulink, we illustrated three graphs the arm angle, pendulum angle, the command graph.

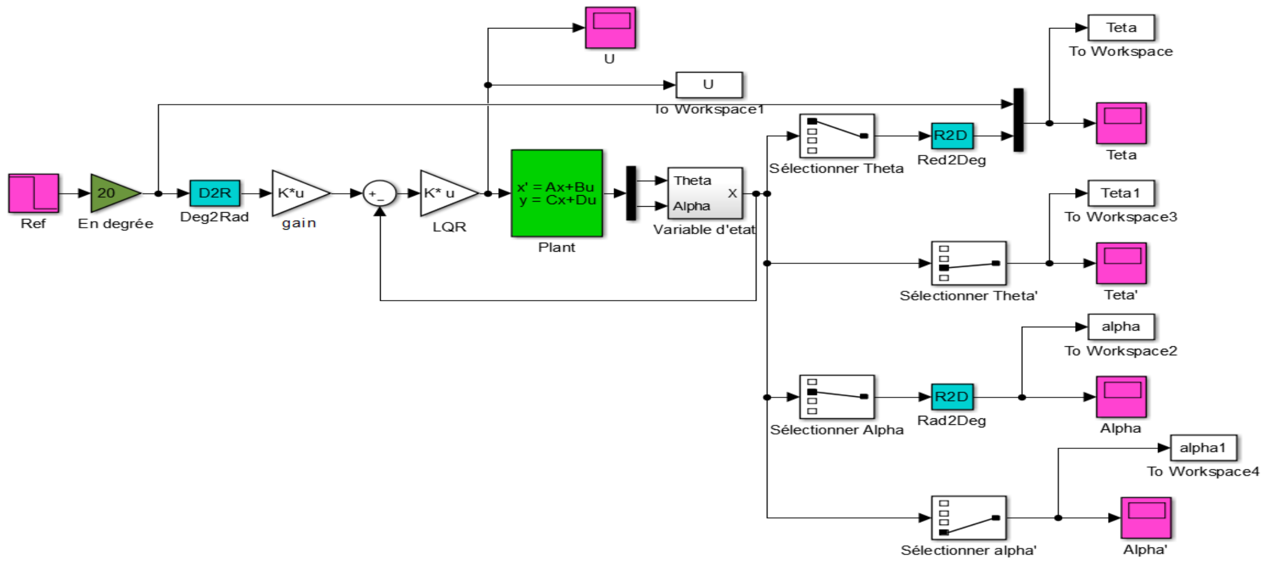


Figure 3-3 The LQR controller implemented on Simulink model of QUBE-Servo

1.22 Results and comparison

The results obtained from the simulation of the two controls applied the LQR controller and the LQR optimized by GAs on our system, For the input of the system, we change the position of the arm θ with a square signal between 20º and -20º degrees. the simulation time is chosen as t_sim=100s, and the sampling time t=0.02s.

which shows the evolution of the arm angles θ and the pendulum angles α and the commands graph, which we illustrated bellow:

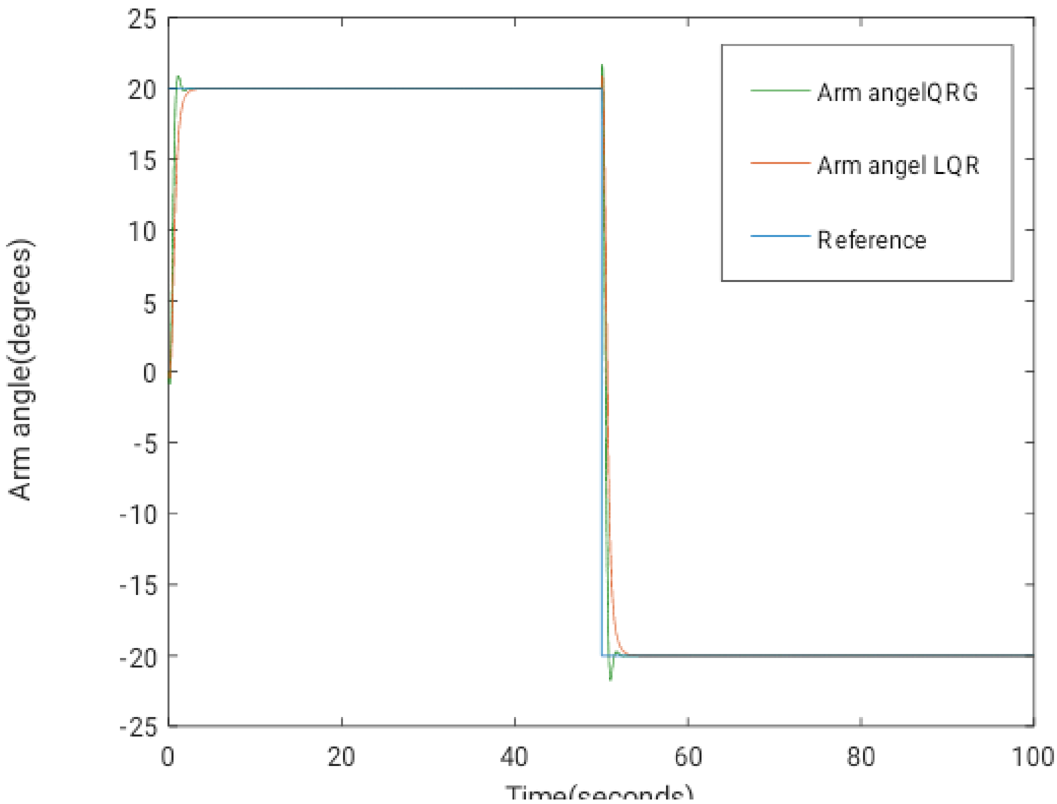


Figure 3-4 The arm angles for two controls LQR and LQR GAs

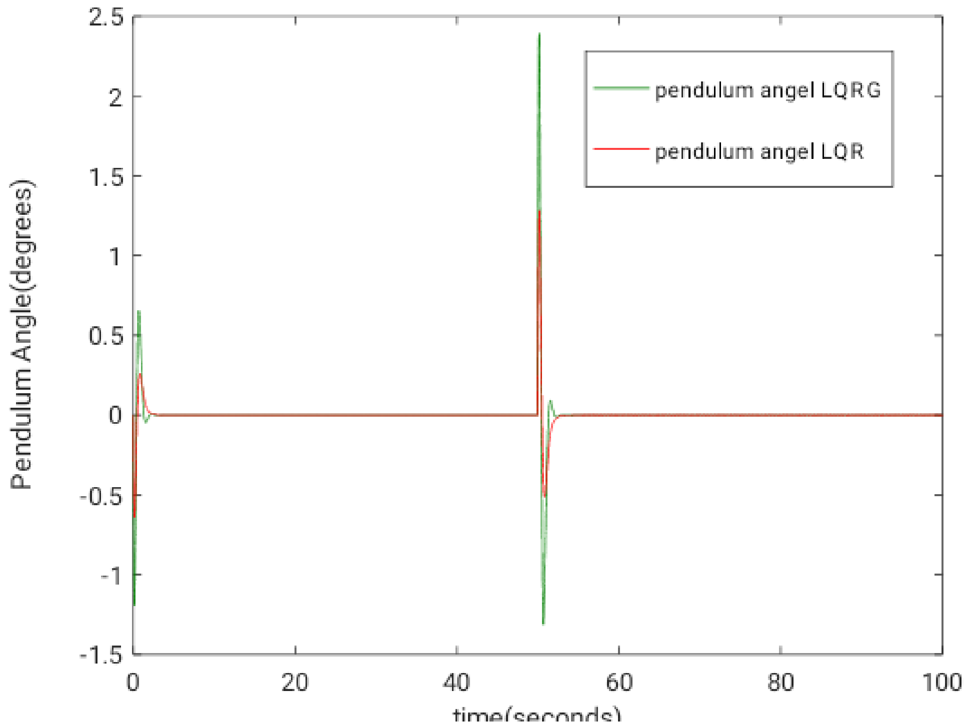


Figure 3-5 The pendulum angles for the two controls LQR and LQR GAs

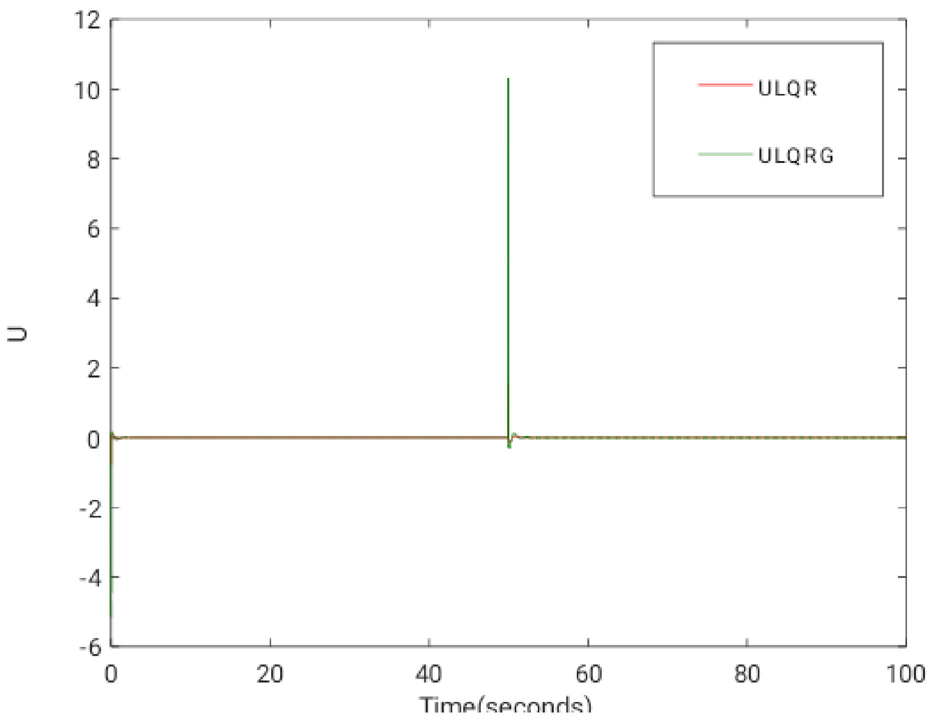


Figure 3-6 The two controls graphs

The results illustrated in the previous curves (3-4….3-6) show that the LQR genetic algorithm control has better performance compared to the LQR control. The arm which is characterized by the angle θ welcoming to the desired position in a shorter time, also stabilizing the pendulum is faster and less oscillating.

1.23 Conclusion

In this chapter, we applied the two commands previously studied (the quadratic linear control and the quadratic linear control genetic optimized by the genetic algorithm) on a pendulum rotary inverted, in order to make the comparison between them. After having our results, we noted that the best order which gives us the least response time and least rise time is the command LQR genetic algorithm.

Chapter IV

Experimental implementation of the controllers on the QUBE-SERVO 2 model

Chapter 4: Experimental implementation of the controllers on the QUBE-SERVO 2 model

1.24 Presentation of the Rotary Inverted Pendulum

The RIP considered in the context of this work is of the Quanser QUBE-Servo 2 type available at level of our establishment, called the Furuta pendulum and is a classic automatic system. It is widely used for testing command types because it represents a nonlinear system and under actuated multivariable.

The Rotary Inverted Pendulum experiment is ideal for studying intermediate to advanced concepts encountered in any system that requires vertical stabilization, from Segway vehicles to rocket launching systems. Equipped with high quality direct drive brushed DC motor, single encoder, internal data acquisition system and amplifier; the Rotary Inverted Pendulum module consists of an arm that mounts to the Rotary Servo Base Unit. The pendulum rod is attached to the arm's metal shaft, instrumented with a high-resolution encoder measuring the pendulum's angle allows us to connect the unit to a PC via USB, an NI myRIO on board device, and other microcontrollers such as an Arduino or a Raspberry Pi using the SPI protocol. The Rotary Servo Base Unit rotates the arm with the pendulum in the horizontal plane. We learn to design controllers that balance the pendulum in the upright position by rotating or changing the angle at the base (inverted pendulum), or swing up the pendulum and maintain it in the upright position (self-erecting inverted pendulum).

Figure 4-1 the Quanser rotating pendulum [1].

1.25 Components of the Rotary Inverted Pendulum

The QUANSER type RIP is composed of an SVR02 motor, which drives an arm with its

end a pendulum (rod) bind to each other by a pivot connection. This system has two sensors,

one encoder wheel each. the different elements of the pendulum are described by the figure 4-2

Figure 4-2 the Quanser rotating pendulum Components [1].

The rotating inverted pendulum module is attached to the SRV02 load gear by two thumbscrews.

The pendulum arm is fixed to the body of the module by a set screw. The RIP experience is A classic example of how the use of the controller can be used to stabilize a system. The inverted pendulum is also an accurate model in the pitch and yaw of a rocket in flight and can be used as a reference for many control methodologies.

1.26 Linear model of RIP

state presentation of the QUBE-Servo rotary pendulum will be changed from our state representation (The values of matrices A and B will be a bit efferent from our matrices)

$$A = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 143.2751\ -0.0109\ \ 0\ 0\ 258.6091\ -0.0107\ 0\ ] \quad (4\text{-}1)$$

$$B = [0\ 0\ 48.7275\ 48.1493\ ]$$

$$C = [1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ ]$$

$$D = [0\ 0\ ]$$

1.27 Implementation of controller on model

To implement the linear controls on the RIP, the pendulum must be manually placed around from its position of unstable equilibrium (area of linearity). Command parameter values used are readjusted in relation to the system of the model.

1.27.1 LQR controller

Using LQR function with loaded model and weight matrices

$$Q = \begin{bmatrix} 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0 \\ 0\ 0\ 1\ 0 \\ 0\ 0\ 0\ 1 \end{bmatrix}, R = 1 \quad (4\text{-}2)$$

from simulation results we constate that the values of Q and R shown below are the best choice for satisfactory commands

$$Q = \begin{bmatrix} 5\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,20 \end{bmatrix}, R = 1 \tag{4-3}$$

With gain matrix:

$$K = [-2.2361 \quad 53.8606 \quad -1.6975 \quad 6.5259] \tag{4-4}$$

We illustrate the graphs of the arm and pendulum angel and commands in next section

1.27.2 LQR genetic controller

For an optimized controller result we use the values of the weighting matrices with the optimal result from the simulation to implement it in real time simulation

$$Q = \begin{bmatrix} 20.5213 \;\; 0\;0\;0 \\ \mathbf{0} \quad 8.6780\;0\;0 \\ 0 \quad 9.2072 \quad 0\;0 \\ 0 \;\; \mathbf{0} \;\; 0 \quad 0.5745 \end{bmatrix}, R = 0.1973 \tag{4-5}$$

With the gain matrix as follow:

$$K = [-14.7979 \;\; 178.4434 \quad -6.8001 \quad 22.3600 \qquad \textbf{(4-6)}$$

We illustrate the graphs of the arm and pendulum angel and commands in next section

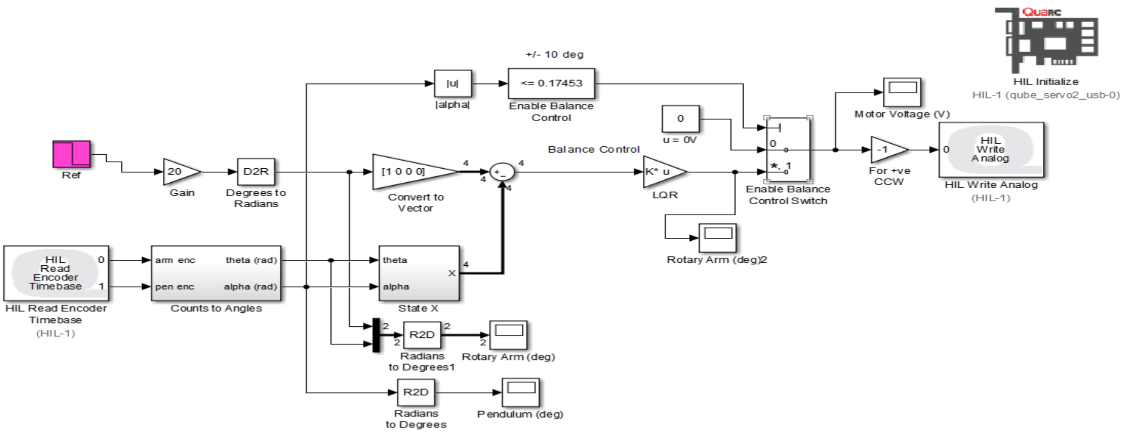


Figure 4-3 real time simulation of LQR controller implemented on model of QUBE-Servo

1.28 Result of the test on the model

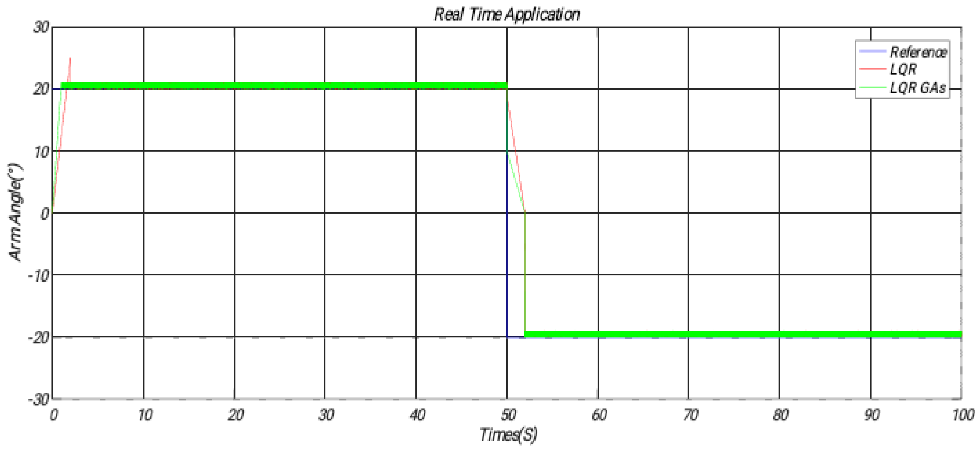


Figure 4-4 The arm angles response for the two controls LQR and LQR GAs

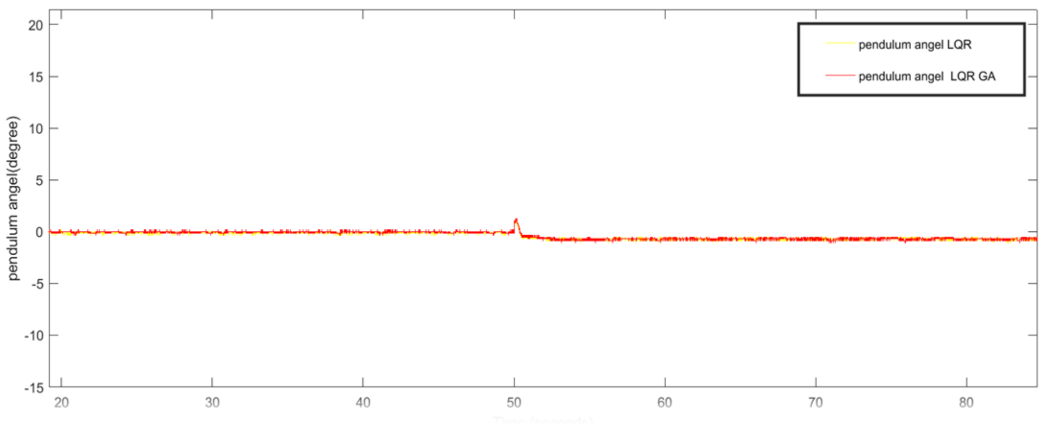


Figure 4-5 The pendulum angles response for the two controls LQR and LQR GAs

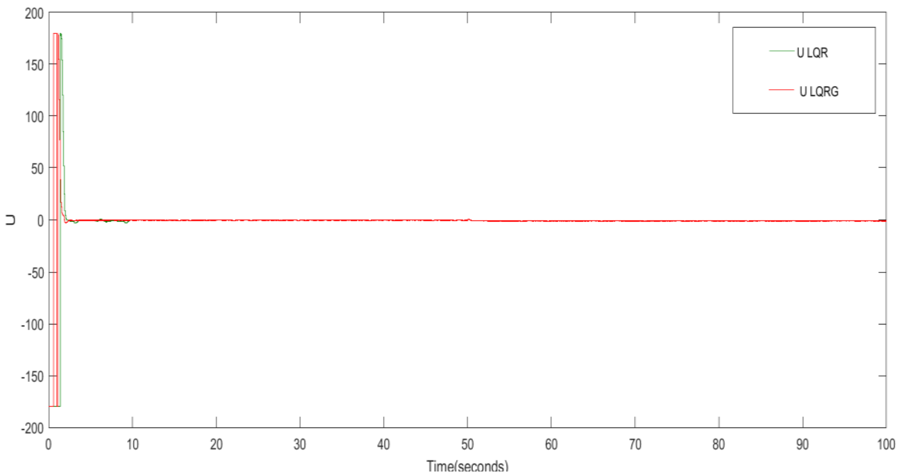


Figure 4-6 The evolution of controls for the two commands LQR and LQR GAs

1.29 Discussions

From the results obtained in the figure (4-4,4-5) we note that that the LQR genetic controller reaches the set point quickly compared to LQR controller, as we can see from these graphs, LQR genetic controller shows better performance compared to the LQR- controller, the arm, which characterized by the welcoming angle θ that has the desired position in a shorter time, also the stabilization of the pendulum is faster and less oscillating.

LQR genetic controller is also characterized by a reduced overshoot and short delay time. In summary, for dynamic response, the inverted pendulum controlled by LQR optimized controller balances faster because of the shorter settling time; and it has better robustness because of the less maximum overshoot. The above points substantiate for the fact that the LQR controller can guarantee the rotary inverted pendulum system a better performance dynamic than a LQR controller.

1.30 Conclusion

In this chapter, we applied two commands of the real-time linear controls (the linear quadratic regulator and the linear quadratic regulator) which are optimized by the genetic algorithm for the RIP. We noticed that the two commands succeeded in stabilizing the RIP even in the presence of disturbances. We found that the optimization by genetic algorithms of the linear quadratic regulator gave the best results in terms of energy efficiency, robustness, and stabilization of the pendulum.

General conclusion

The work presented in this project focuses mainly on modeling the LQR command to stabilize the rotary inverted pendulum that is considered as a highly unstable non-linear system. The objective is to stabilize the rotary inverted pendulum around its unstable equilibrium point. We used the Euler-Lagrange formalism to model the system mathematically; but with its complex nature, it became very difficult to command, then we proceeded by the linearization of this system; then, we synthesized linear controller; Quadratic Linear Regulator (LQR) and the optimized LQR using genetic algorithm in order to see its impact on controlling the non-linear system with two degrees of freedom. We applied the controller in real time on the model of Quancer type pendulum. Finally, the experimental results show the efficiency of the proposed controller.

Bibliography

[1]. Documents Quenser. QUBE-SERVO 2 Workbook - Student. www.quanser.com.

[2].Kerroum, Fatima Zohra and Benarouba, Naouia, Commande d'un pendule rotatif inversé par des techniques optimises, Mémoire présenté en vue de l'obtention du diplôme de MASTER, Université de Ghardaïa, 2021

[3] .Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum. Kumar, E Vinodh and Jerome, Jovitha. 2013. 2013, Procedia Engineering, pp. 169--178.

[4]. Naidu, D Subbaram. 2002. {Optimal control systems},. {CRC press} : s.n., 2002.

[5]. ow cost two-wheels self-balancing robot for control education. Gonzalez, C and Alvarado, I and La Pe{\~n}a, D Mu{\~n}oz. 2017. s.l. : Elsevier, 2017.

[6]. Alexis Delisse and Christophe Le Mao. Moto autonome : Honda riding assist-e, Oct 2017.

[7]. Khaoula BRAHIM. Analyse et modélisation : du pendule inversé sur deux roues au fauteuil roulant automatisé freemove». Rapport de Stage de Master, 2

[8]. ADDAOUD Hocine, Étude comparative d'un contrôleur floue basé sur LQR et Un contrôleur de mode glissant Application sur un pendule inversé rotatif , Mémoire présenté en vue de l'obtention du diplôme de MASTER, Université de Ghardaïa, 2020

[9]. On the linear control of underactuated systems: The flywheel inverted pendulum. Olivares, Manuel and Albertos, Pedro. 2013. s.l. : IEEE, 2013.

[10].H. Ludvigsen A. Shiriaev, A. Pogromsky and O. Egeland. On global properties of passivity-based control of an inverted pendulum. « International Journal Of Robust And Nonlinear Control », (10) :283–300, 2000

[11].W. Zhong and H. Röck. « Energy and passivity based control of the double inverted pendulum on a cart ». In Proceedings of the 2001 IEEE international conference on control applications, Mexico City, Mexico, 2001.

[12]. Triantafyllou, Michael S and Hover, Franz S and others. 2003. vehicles, Maneuvering and control of marine. s.l. : Citeseer, 2003

[13]. Franklin, Gene F and Powell, J David and Emami-Naeini, Abbas and Powell, J David. 2002. Feedback control of dynamic systems. 2002.

[14]. Wongsathan, Chaiporn and Sirima, Chanapoom. 2009. Application of GA to design LQR controller for an inverted pendulum system. IEEE, 2009.

[15]. Sivanandam, SN and Deepa, SN. 2008. Genetic algorithms. Introduction to genetic algorithms. s.l. : Springer, 2008.

[16].Yu, Wei and Li, Jie and Yuan, Jing and Ji, Xi. 2021. LQR controller design of active suspension based on genetic algorithm. IEEE, 2021.

[17]. Beg, Abul Hashem and Islam, Md Zahidul. 2016. Advantages and limitations of genetic algorithms for clustering records. IEEE, 2016.

[18]. Pham, Duc and Karaboga, Dervis. Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks. 2012 : Springer Science \& Business Media.

[19]. Sastry, Kumara and Goldberg, David and Kendall, Graham. 2005. Genetic algorithms. *Search methodologies.* s.l. : Springer, 2005.

[20]. algorithms, Evolutionary computation: theory and. Nayyar, Anand and Garg, Surbhi and Gupta, Deepak and Khanna, Ashish. *Advances in swarm intelligence for optimizing problems in computer science.* 2018 : Chapman and Hall/CRC.

[21]. Deb, K. Multi-objective optimization using evolutionary algorithms (2005)

[22]. *Parameter optimization of linear quadratic controller based on genetic algorithm.* Li, Jimin and Shang, Chaoxuan and Zou, Minghu. 2007. s.l. : TUP, 2007.

[23]. *Genetic algorithm: Review and application.* Kumar, Manoj and Husain, Mohammad and Upreti, Naveen and Gupta, Deepti. 2010.

[24]. *Pymoo: Multi-objective optimization in python.* Blank, Julian and Deb, Kalyanmoy. 2020. s.l. : IEEE, 2020.

[25]. Pillai, Anju G and Samuel, Elizabeth Rita and Unnikrishnan, A. 2018. Analysis of Optimised LQR Controller Using Genetic Algorithm for Isolated Power System. s.l. : Springer, 2018.

[26]. Gu, Jingming and Fang, Dianjun. 2021. Genetic Algorithm Based LQR Control for AGV Path Tracking Problem. *Journal of Physics: Conference Series.* 2021.

[27]. Fang, Hui and Tang, Wei and Zheng, Enrang. Research on inverted pendulum control based on LQR. *Advances in Mechanical and Electronic Engineering.* 2012 : Springer.

[28]. Boubaker, Olfa and Iriarte, Rafael. 2017. *The inverted pendulum in control theory and robotics: from theory to new innovations.* s.l. : IET, 2017.

[29]. BERKOUNE, Ali Abdessetar and RABEHI, Mohammed Rida, Modélisation et commande en temps réel d'un pendule inversé rotatif: Approche linéaire., Mémoire présenté en vue de l'obtention du diplôme de MASTER, 2021

[30]. Kathpal, Abhishek and Singla, Ashish. 2017. SimMechanics™ based modeling, simulation and real-time control of Rotary Inverted Pendulum. *2017 11th International Conference on Intelligent Systems and Control (ISCO).* IEEE, 2017.