2DOF ROBOTIC ARM

BACHELOR’S PROJECT OCT2022

PID-Controller Tuning Optimization with Genetic Algorithms for Robotic Arm

Mahsa Seraji
Dr. Tahereh Binazadeh

TABLE OF CONTENTS

Contents

ABSTRACT & INTRODUCTION

Abstract & Introduction

ABSTRACT

Robotics is one of the important branches of control science and has numerous control applications. In the field of robotics, we deal with nonlinear modeling, as our systems have nonlinear structures. Researchers study the design of linear controllers for these systems to achieve optimal behavior and performance. In this project, we aim to combine linear controller structures with intelligent structures such as genetic algorithms to design the best coefficients for these structures. This allows us to achieve the best performance in simulation of the designed controller and ultimately demonstrate the effectiveness of the control method based on conducting simulations on the system.

INTRODUCTION

A robot arm typically consists of multiple joints, and due to their widespread application, controlling them is crucial. PID controllers are widely used for controlling the speed and position of robot arms. However, obtaining PID coefficients poses challenges. To address this issue, we employ intelligent control methods, which will be fully discussed in the following sections. In this project, we aim to design a system using a genetic algorithm.

The genetic algorithm is a random algorithm based on principles of natural selection and genetics, simulating the natural evolution process. Using this algorithm for controller tuning allows for the evaluation of the optimal controller for the system each time.

In this project, we intend to achieve optimization using the genetic algorithm and find the best coefficients for the controller. We will further explore robot control, its industrial applications, dynamic modeling of the robot arm and its equations, intelligent design methods, controller design, and simulation.

Robot Control and its Applications in Industry

1-2. Robotic

At the outset, to understand how robot control works, we delve into the field of robotics, a branch of technology concerned with the design, construction, operation, and application of robots and computer systems for control and more.

Robotics is essentially an interdisciplinary field combining science, engineering, and technology dedicated to the design, construction, and use of mechanical robots.

1-3. Robots

A robot is a programmable machine capable of completing a task, while robotics describes the academic discipline focused on the development of robots and automation. Each robot has a different level of anatomy. These levels may range from human-controlled bots to fully automatic robots that perform tasks without any external influence.

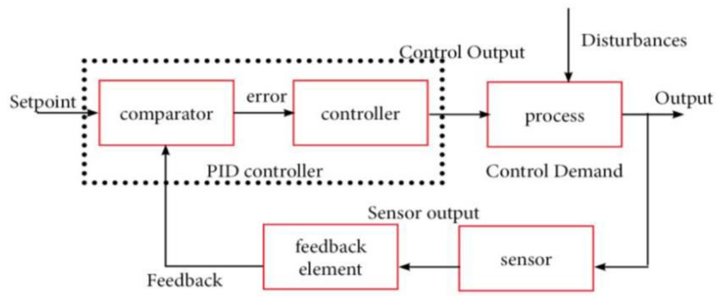
1-4. Control Systems in Robotics

Every robotic system is manufactured to perform a required function. Control systems enable the movement and operation of various parts of the robot, and in case of unpredictable errors, they execute a series of movements and forces.

1-5. Different Types of Control Systems:

1-5-1. Control using PID:
PID controllers are used in over 90% of industrial control applications, mostly because they have multiple tuning rules, and most of these controllers can be tuned using these rules. Some of these controllers even have the capability of automatic and online tuning. Significant advancements have been made in the field of Artificial Neural Networks (ANN), Fuzzy Logic (FL), Artificial Intelligence (AI), and evolutionary computational methods such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and more.

All of these techniques provide new solutions for solving problems in various control systems. For example, for controlling a two-wheel self-balancing robot, a PID controller is suitable.



1-5-2. Control using LQR:

LQR, or Linear Quadratic Regulator, is a modern feedback control method that utilizes a state-space approach to analyze and design systems. LQR is used in combination with PID and FLC to assist in controlling a two-wheel self-balancing robot.

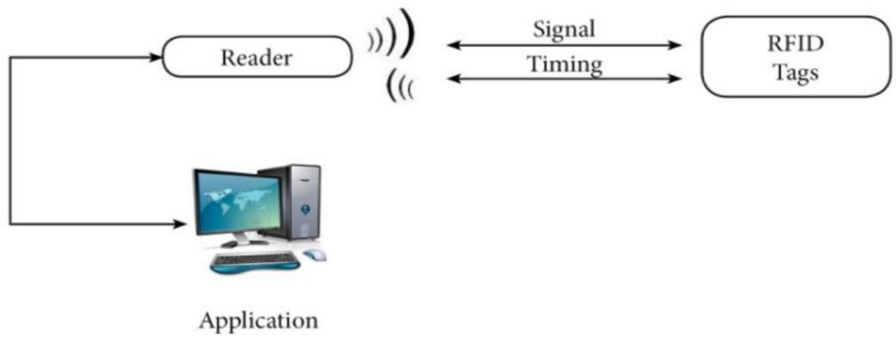
In essence, LQR is a procedure in modern control where the state-space method is employed to examine the system. The LQR control method is ideal for the overall design approach of Multi-Input Multi-Output (MIMO) systems.

It considers the dynamic framework conditions and control information to establish ideal control choices.

1-5-3. Control using RFID:

RFID, or Radio Frequency Identification, enables wireless access to system control and consists of three components:
1. RFID tag
2. Reader for the tag
3. Computing device for managing data information

The application of RFID is in military and security applications. In the figure below, you can see the reader and tag, which are the main components of an RFID system:



In other articles, various types of robot control systems have been analyzed as follows:

- **Point-to-Point Robot Control System:**
  - Used to determine a location where a piece can be picked up or released.
  - Operates during loading and unloading of programs.

- **Continuous Path Robot Control System:**
  - Used for situations where the robot needs to follow a specific path, such as welding or painting.

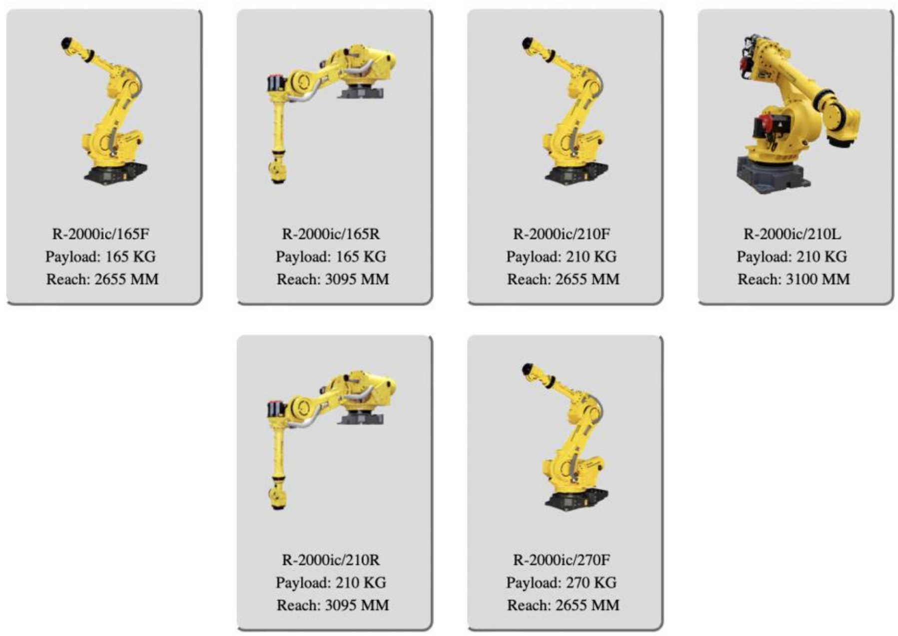
Additionally, robots can be controlled through various means including manual control, wireless control, and fully automated control using artificial intelligence.

1-6. Applications of Robot Controllers in Industry:

1-6-1. Robot controller is one of the key components of an industrial robot. It is a computer system that is connected to the industrial robot's arm to control its movements. In addition to controlling the robot arm, the controller is also responsible for stopping and preventing interference in the robot's workspace. All industrial robots are paired with a controller to function properly.

For example, the FANUC Lr Mate 200id is paired with the R-30ib controller, while the FANUC LR Mate 200ic is paired with the R30ia controller.

FANUC LR Mate 200ID



1-6-2. Robot Controller is often regarded as the brain of a robot because it interprets the code, known as programming, for a specific robot. The controller deciphers the codes into instructions for the robot to use for its functioning and executing the program steps. Robotic programs are encoded through teach pendants on the controller.

These teach pendants are a critical component of the control system as they serve as the primary method for programming the robot and are composed of buttons and touch-screen displays to facilitate the input of programming commands.

You can see several different examples of teach pendants in the image below:

ABB
Teach Pendants

FANUC
Teach Pendants

KUKA
Teach Pendants

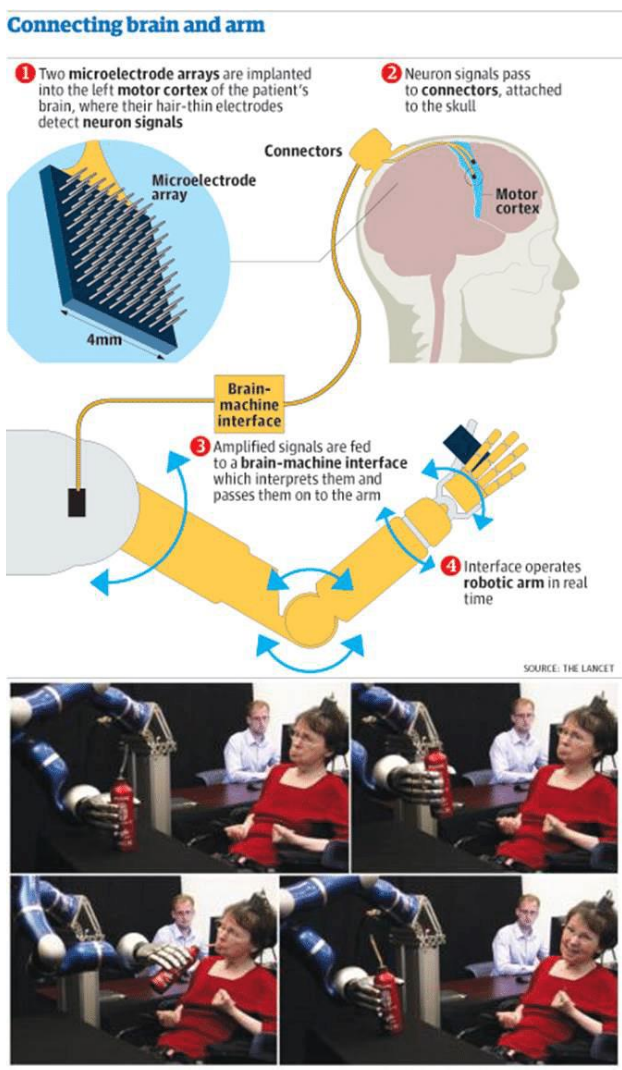
Yaskawa
Teach Pendants

1-6-3. Example of Commonly Used Controller Models: Yaskawa Motoman XRC Controller

The Yaskawa Motoman XRC Controller is an industrial robot controller that features user-friendly software and a Teach pendant with Windows support. In 2001, the XRC control was updated to enhance safety standards, including dual-channel emergency stop. The Yaskawa XRC is capable of controlling up to 4 robots using just one teach pendant, reducing the cost of multi-robot systems like Arcword 6200. These robots have widespread applications due to their durability.

1-6-4. Another example of the application of controllers in robots is a robotic arm controlled by the human brain. Due to the high cost and risks associated with implanting sensors in the brain, researchers have sought ways to control robotic arms through non-invasive or even completely indirect methods, allowing paralyzed patients to use them to control their surrounding environment without the need for surgery. When controlling a robotic arm using the brain, Brain-Computer Interface (BCI) technology does not rely on implanted devices inside the body.

This technology can serve a wide range of people by providing safe and non-invasive mind control over tools that assist them in interacting with their environment.

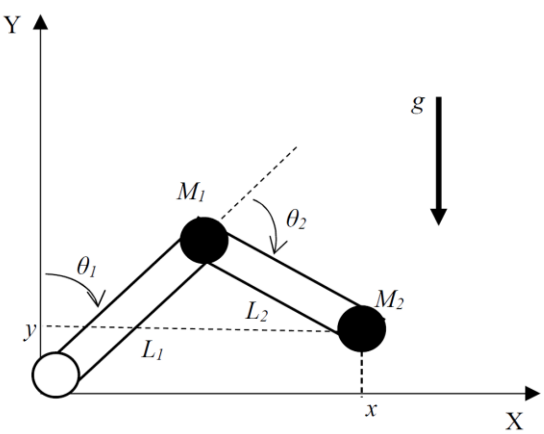


Dynamic Modeling of a 2DOF Robotic Arm

In all robot applications, the execution of specific tasks by the robot leads to the control of the end effector along the desired path. Controlling the position of the end effector requires precise analysis of the mechanical structure characteristics of the robot arm and actuators. The goal of this analysis is to achieve a mathematical model of the robotic system.

Extracting the dynamic model of the robotic arm and actuation system plays a crucial role in analyzing the structure and motion of the arm and designing the control system.

A robotic arm with two degrees of freedom is depicted in Figure 1, where θ_i ، L_i & M_i $\{i = 1,2\}$ represent the joint angles, lengths and masses of the first $\{i = 1\}$ and second $\{i = 2\}$ links, respectively. The gravitational force is indicated by g.



1. Two-Degree-of-Freedom Robotic A

The dynamic model of this robot is calculated based on its kinetic and potential energies, using a direct geometric model with the following formula:

1. $$\begin{cases} x = L_1 \sin(\theta_{1)} + L_2 \sin(\theta_1 + \theta_2) \\ y = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \end{cases}$$

DYNAMIC MODELING OF A 2DOF ROBOTIC ARM

Using equation 1, the total kinetic energy of the two-degree-of-freedom robotic arm is obtained with the following equation:

2.

$$E = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_1^2 + M_2L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\left(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2\right)\cos(\theta_2)$$

And the potential energy is obtained with the following formula:

3.

$$U = M_1gL_1\cos\theta_1 + M_2g(L_1\cos\theta_1 + L_2\cos(\theta_1 + \theta_2))$$

To derive the equations of motion for the robot, we use the Lagrange equation:

4.

$$L = E - U$$

With the Lagrangian L , we can solve the Euler-Lagrange equation, which relies on partial derivatives of the kinetic and potential energy properties of mechanical systems to calculate the equations of motion. It is defined as follows:

5.

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}_i}\right) - \frac{\partial L}{\partial\theta_i}$$

Where L and $\tau = [\tau_1 + \tau_2]$ represent the Lagrangian of motion and the torque vector, respectively. By expanding equation (5), the dynamic model of a two-degree-of-freedom (DOF) robotic arm is represented by the following formula:

6.

$$\begin{cases} M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) = \tau \\ Y = \theta \end{cases}$$

Which,

The joint angle vector: $\theta = [\theta_1, \theta_2]$

$\tau = [\tau_1, \tau_2]$: The torque vector (control input)

Y: The output vectors

$G(\theta)$: The vector of gravitational torques, obtained from the following equation:

7.

$$G(\theta) = \begin{bmatrix} -(M_1 + M_2)gL_1 \sin(\theta_1) - M_2 g L_2 \sin(\theta_1 + \theta_2) \\ -M_2 g L_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$C(\theta, \dot{\theta})$: The vector of Coriolis and centrifugal forces, obtained from the following equation:

8.

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -M_2 L_1 L_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_1^2\right) \sin \theta_2 \\ -M_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 \end{bmatrix}$$

$M(\theta)$: The inertia matrix is represented as:

9.

$$M(\theta) = \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix}$$

In which:

$$D_1 = (M_1 + M_2)L_1^2 + M_2 L_2^2 + 2M_2 L_1 L_2 \cos(\theta_2)$$

$$D_2 = M_2 L_2^2 + M_2 L_1 L_2 \cos(\theta_2)$$

$$D_3 = D_2$$

$$D_4 = M_2 L_2^2$$

In the coding section, the application of these formulas is demonstrated.

Intelligent Design Methods

Intelligent control is a category of control techniques that utilize artificial intelligence computational methods such as neural networks, fuzzy logic, machine learning, etc. Intelligent control aims to automate tasks by emulating biological intelligence and seeks to replace human control operators (such as a chemical process operator) or to apply ideas from biological problem-solving and implement them for control problem-solving (such as using neural networks for control).

Types of Intelligent Methods:

PID controllers are the most commonly used controllers in industry. Despite their widespread application, they come with challenges, the main one being the determination of appropriate coefficients and constants for these controllers, for which there hasn't been a precise solution yet. The remedy to this problem lies in employing intelligent optimization algorithms to determine the PID controller coefficients.

These algorithms fall into four categories, including:

- Genetic Algorithm (GA)
- Particle Swarm Optimization (PSO)
- Differential Evolution (DE)
- Imperialist Competitive Algorithm (ICA)

Apart from these algorithms, other intelligent control methods have also been mentioned in other articles, including:

- neural network control
- machine learning control
- reinforcement learning control
- Bayesian control
- fuzzy control
- neuro – fuzzy control
- expert systems

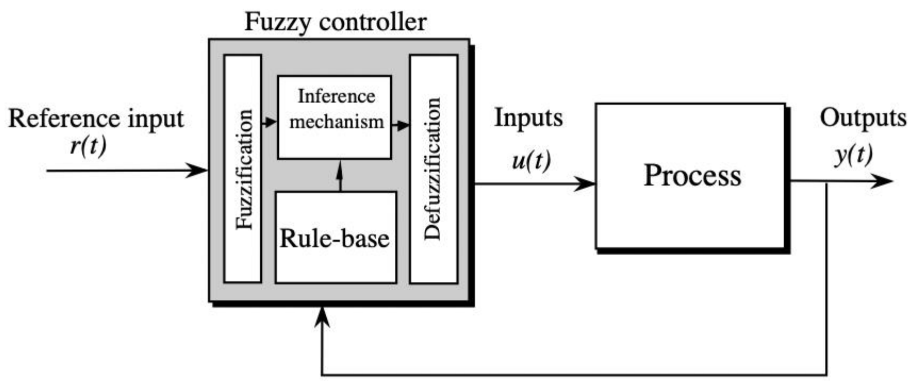
- genetic control

In this section, an attempt has been made to briefly explain some of these intelligent methods:

- **Fuzzy Control:**

Fuzzy control is a method for representing and implementing human knowledge (human intelligence) about how to control a system. In the figure below, a fuzzy controller is depicted, consisting of the following components:

- Rule base: It includes rules governing the control process.
- Fuzzification: It transforms numerical inputs into a form that can be used by the inference mechanism.
- Inference mechanism: It utilizes information about the inputs (generated by fuzzification) to decide which rules to apply under current conditions and determines which input should be the plant.
- Defuzzification: It converts the results obtained from the inference mechanism into a numerical input for the plant.



Challenges in Fuzzy Controller Design:

- A deep understanding of control problems, including plant dynamics and loop characteristics, is required.
- The rule base must be carefully constructed, or else it may lead to failure.
- In practical applications, complex issues may arise because the number of rules used, especially when using rule combinations, increases exponentially with the number of controller inputs.

INTELLIGENT DESIGN METHODS

- **Artificial Neural Networks (ANNs):**

Artificial neural networks are circuits, computational algorithms, or mathematical representations consisting of interconnected neurons that mimic biological neural networks. ANNs are a computational technology that is useful for various pattern recognition, signal processing, estimation, and control problems.

Challenges in ANN Design:

- Selecting a training set (G) that can guarantee good approximation is challenging.
- Choosing an appropriate structural approximation is difficult.
- Generally, due to the presence of many local minima, ensuring convergence of training methods is impossible, meaning it's hard to determine when the algorithm has finished.
- There is a significant issue with generalization, where we hope the neural network can interpolate well between similar training inputs, and ensuring good interpolation is difficult.

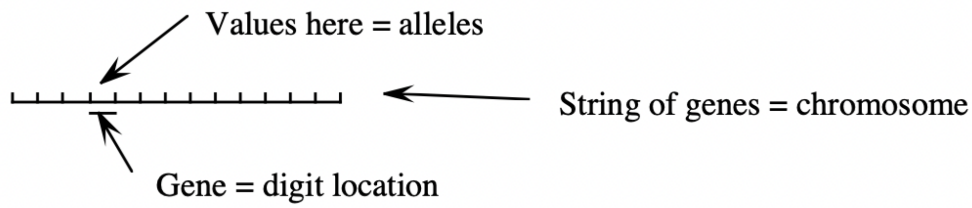
- **Genetic Algorithm**

The genetic algorithm is a computer program that simulates the principles of evolution, natural selection (Darwinism), and genetics. It's an optimization search technique based on selected natural mechanisms. This technique starts with an initial population consisting of a number of chromosomes, each representing a solution to a problem, and has three stages:

- Selection
- Crossover
- Mutation

Initially, a random population is created and evaluated. Then, parents are selected for mutation and combined, resulting in a mutated population. Finally, the original population, offspring, and mutated individuals are merged, creating a new original population. This process is repeated until termination conditions are met. Termination conditions include reaching an acceptable level of solution, elapsed time, or a certain number of iterations without observing significant improvement in the result. Ultimately, this iteration stops when individuals representing the optimal solution to the problem are reached.

Population Individuals:
The fitness function of the genetic algorithm measures the quality of the optimization solution (e.g., biologically, the survival capability of an individual). The algorithm seeks to maximize the fitness function $j(\theta)$ by selecting individuals indicated by parameter θ. According to the figure below, in the genetic algorithm, θ is a string called a chromosome:



To configure an artificial network, a chromosome can be used, which is a combination of the weights and biases of the network. For tuning a PID controller, the chromosome is a combination of three controller gains.

According to Darwin's theory, individuals who are better adapted survive for mating, indicated by $J(\theta^j(k))$, and this mating space is as follows:

$$M(K) = \left\{M^j(K): j = 1, \dots, S\right\}$$

Now if:

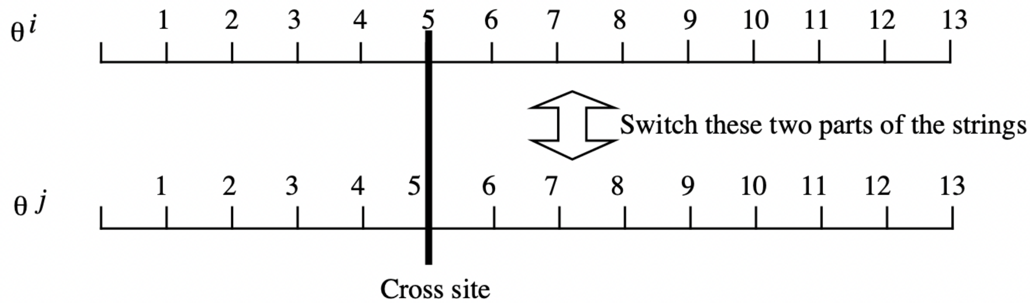
$$M^j(K) = \theta^i(K)$$

Only if:

$$P_i = \frac{J\left(\theta^i(K)\right)}{\sum_{j=1}^{S} J\left(\theta^j(K)\right)}$$

With this approach, individuals with better fitness are more likely to mate, and vice versa.

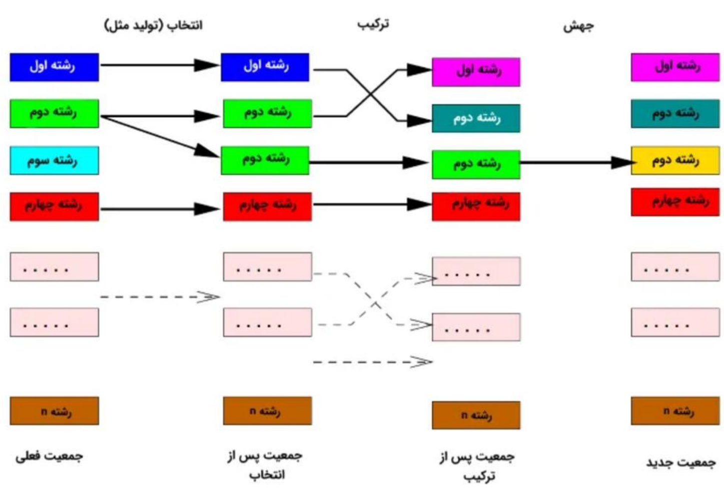
Now that the selection stage is complete, we move on to the crossover and mutation stages. The crossover process, biologically speaking, is equivalent to mating, where individuals are randomly paired with each other. This process is illustrated in the diagram below:



For this, a crossover point is considered, and all strings are directed to the right.

The next stage is mutation, where random genetic changes occur. With a probability of gene change, the gene position in each chromosome randomly changes to a member of the number system used. Mutation ensures that we don't get stuck at a local maximum of the fitness function. Since mutation is a purely random search, usually, the probability P_m is close to zero. Finally, the next generation is produced in this way:

$$P(K+1) = M(K)$$

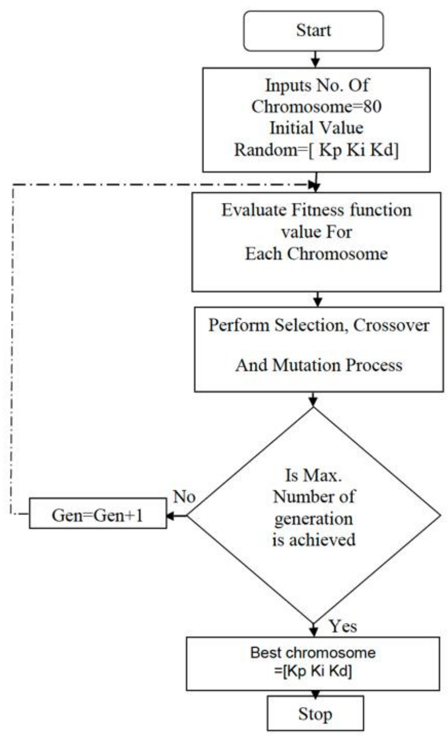


نمای کلی از فرایند تکاملی در الگوریتم ژنتیک پس از تولید مثل، ترکیب و جهش

Optimization in Genetic Algorithm:

For optimizing the objective function in the genetic algorithm, we may use minimization or maximization. If minimization is used, our objective function includes cost and error functions, and if maximization is used, our objective function includes fitness and profit functions.

In the figure below, to better understand the computations of PID/GA controller parameters, its flowchart has been calculated:



Flowchart of Parameter Computations

INTELLIGENT DESIGN METHODS

Challenges of Genetic Algorithm Intelligence Method:

- A comprehensive understanding of optimization is crucial in this area.
- Excessive detail in representation leads to computational complexity.
- There is a wide range of other genetic operators, and selecting the most suitable ones is critical as it affects convergence, such as elitism, where the best individual is passed to the next generation without being disturbed by crossover or mutation.
- Selecting a good termination criterion is important.
- Guaranteeing convergence in practical problems is not feasible.

In this project, we use the intelligent method of the Genetic Algorithm to optimize the PID coefficients for obtaining the best results.

Controller Design

In order to control a robot arm with two degrees of freedom, the structure shown in the figure below is utilized:

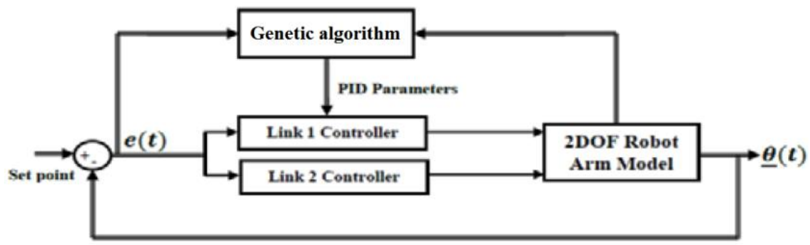


Figure 3. Controller Structure for a Two-Degree-of-Freedom Robot Arm

In the structure shown in Figure 3, the controllers are of PID type, consisting of proportional, integral, and derivative gains of the error signal. In a PID controller, we have:

10.

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt}$$

The current objective is to find suitable coefficients K_P ، K_I & K_D for each joint to minimize joint path errors. The most important aspect in achieving an optimal optimization process is the selection of the objective function used in fitting the candidate solution. A well-formulated fitting function leads to optimal solutions throughout the optimization process. In this robotic system, the objective function for optimizing the controller parameters is formulated as follows:

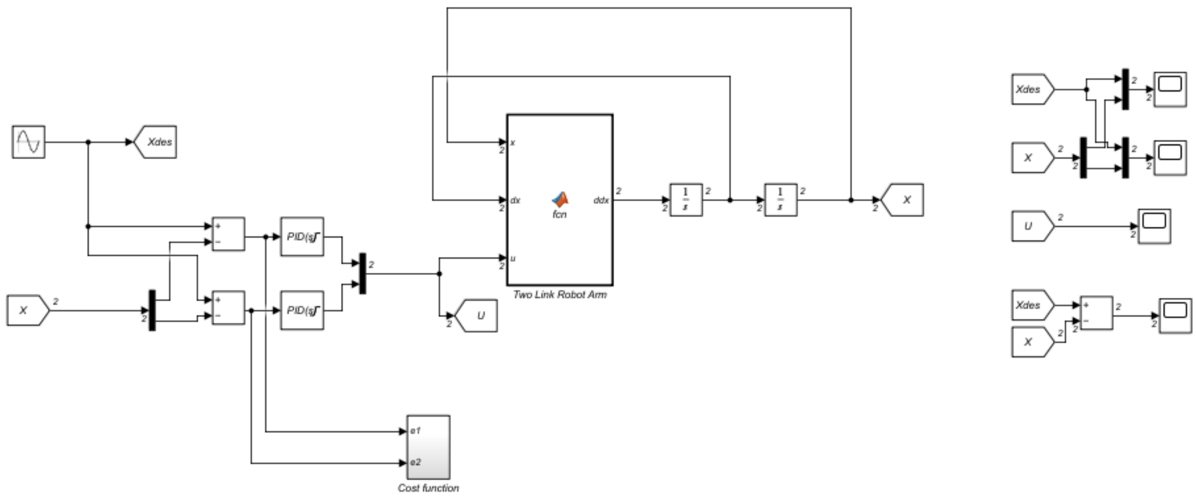
11.

$$J = \int [t|e_{\theta_1}(t)| + t|e_{\theta_2}(t)|]dt$$

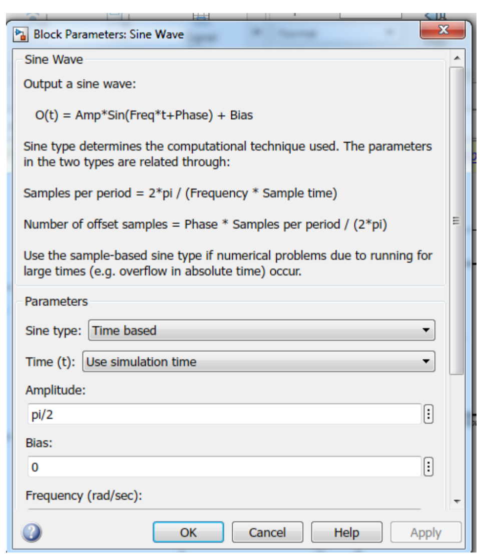
Here, to achieve optimal values for each of the PID controllers, the genetic algorithm is used to minimize the objective function (11).

Simulation

In the MATLAB Simulink section, we have:

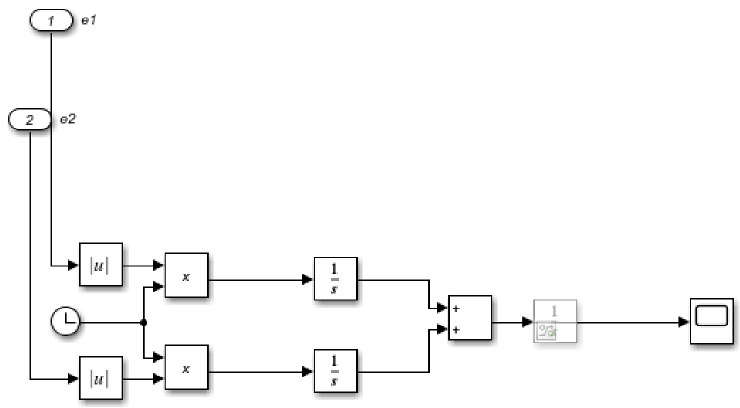


A sinusoidal input reference has been considered.



SIMULATION

The cost function is as follows:



In the robotic arm section, we have:

Robot_arm_2dof ▸ Two Link Robot Arm

```
function ddx = fcn(x,dx,u)

M1 = 1;
M2 = 1;
L1 = 1;
L2 = 1;
g = 9.81;
G = [-(M1+M2)*g*L1*sin(x(1))-M2*g*L2*sin(x(1)+x(2));...
     -M2*g*L2*sin(x(1)+x(2));];
C = [M1*L1*L2*(2*dx(1)*dx(2) + dx(1)^2)*sin(x(2));...
     -M2*L1*L2*dx(1)*dx(2)*sin(x(2));];
M = [-(M1+M2)*L1^2 + M2*L2^2 + 2*M2*L1*L2*cos(x(2)), M2*L2^2 + M2*L1*L2*cos(x(2));...
      M2*L2^2 + M2*L1*L2*cos(x(2)),  M2*L2^2;];
ddx = inv(M)*(-C-G+u);

```

SIMULATION

Results of the simulation:

In this section, to execute the simulation with pre-optimized values or to perform optimization to obtain optimal values, run the "run-me" file.

```
close all

clc

clear all

warning('off')


sel=0; %if sel=0, the optimized gains are used; otherwise, the GA is run for optimizing the PIDs gains.


if sel~=1||sel~=0

    error('The parameter sel is not true')

end

switch sel

    case 0

        x=[-1000.0  -20.58864956120948  -231.25145229133176 785.9660430823861   -383.5336983898735   26.991404536072157];

    case 1


        fitnessfcn=@my_fun;

        LB=ones(1,6)*-1000;

        UB=ones(1,6)*1000;

        nvars=6;

        options = gaoptimset('PlotFcns',{@gaplotbestf},'Display','iter','PopulationSize',50,'InitialPopulation',rand(1,6)*200,'Generations',100);
```

```
        [x,fval] = ga(fitnessfcn,nvars,[],[],[],[],LB,UB,[],options);
end


simopt =
simset('solver','ode45','SrcWorkspace','Current','DstWorkspace','Current');

sim('Robot_arm_2dof.mdl',[0 3],simopt);

plot(x1.time,x1.signals.values)

xlabel('Time(s)')

ylabel('\theta_1(deg)')

legend('Ref','Robot arm')

figure

plot(x2.time,x2.signals.values)

xlabel('Time(s)')

ylabel('\theta_2(deg)')

legend('Ref','Robot arm')
```

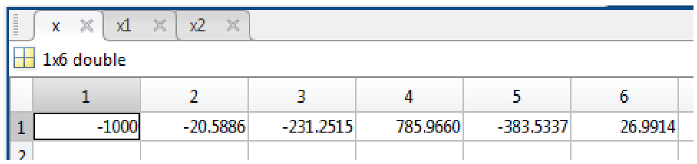
If the defined variable sel is zero, it will use the obtained coefficients, and if it's one, it will execute optimization to obtain new coefficients.

The cost function file is structured as follows:

```
function z = my_fun(~)

simopt =
simset('solver','ode45','SrcWorkspace','Current','DstWorkspace','Current');

 sim('Robot_arm_2dof.mdl',[0 3],simopt);

 OF1=F.signals(1).values;

 OF1_n=OF1;

 z=OF1_n(end);
```

SIMULATION

After optimization, the controller parameters were obtained as follows:



x | x1 | x2

1x6 double

	1	2	3	4	5	6
1	-1000	-20.5886	-231.2515	785.9660	-383.5337	26.9914
2						

4.

We have:

	K_I	K_P	K_D
PID-1	-1000	-20.59	-231.25
PID-2	785.97	-383.53	26.99

The following figures depict the simulation results for the sinusoidal reference signal with a frequency of 1 rad/s. Initial values for the joints were chosen as π/2 and -π/2, respectively. From these figures, it is evident that the optimized PID controller with the genetic algorithm was able to effectively control both joints.

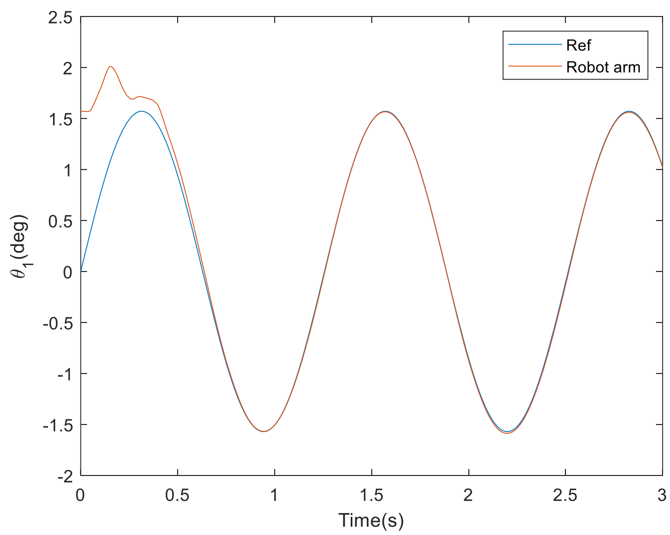


Figure 5. Angle of Joint 1 for Sinusoidal Reference Input

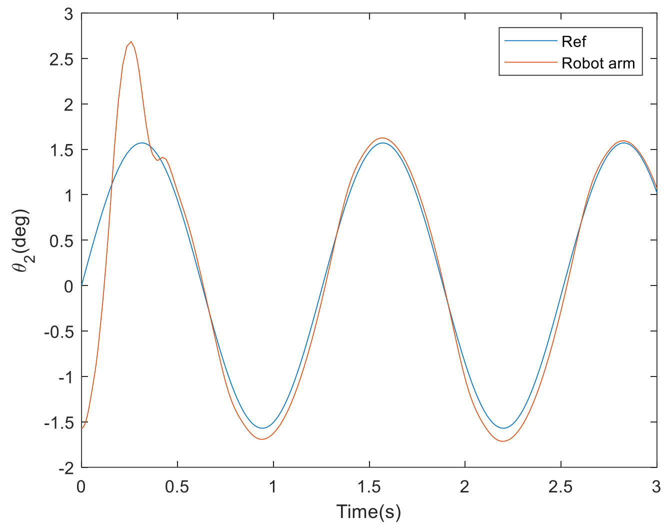


Figure 6. Angle of Joint 2 for Sinusoidal Reference Input