

February 13, 2023

Double Pink Arm With Wrist Controller Design

Javier Irizarry-Delgado
FRC Team 2383, The Ninjineers

Contents

1	Introduction	3
1.1	What is the Pink Arm?	3
1.2	Purpose	3
1.3	Who should read this paper?	3
1.4	What is a double pink arm with a wrist?	3
2	Kinematics	3
2.1	Forward Kinematics	3
2.2	Inverse Kinematics	4
3	Feedforward Control	5
3.1	Introduction	5
3.2	K_v and K_a	5
3.3	K_s	5
3.4	K_g	5
3.5	Conclusion	6
4	PID Control	7
4.1	Introduction	7
4.2	Proportional Gain	7
4.3	Derivative Gain	7
4.4	Integral Gain	8
5	Feedforward and PID Control	8
5.1	Putting it all together	8
6	Motion Profiling	9
7	Presets	9

1 Introduction

1.1 What is the Pink Arm?

The Pink Arm is an arm design originally used by Team 233 (The Pink Team) in 2004 to play defense and climb in endgame. It is comprised of a single jointed arm, with a telescoping joint at the end. This is a very versatile design, as it allows a subsystem mounted at the end to span a large area. This is useful for manipulating game elements, and has proven over the years to be a very efficient design in FRC.

1.2 Purpose

The purpose of this paper is to explain how concepts like PID control, feedforwards, forward and inverse kinematics, motion profiling, and presets, play a role in the programming of a complex robot subsystem. These concepts can turn a mechanism into a competitively viable subsystem that is easy to control in the teleoperated and autonomous periods of the FRC match.

1.3 Who should read this paper?

This paper is aimed at high school students in FRC who are looking to learn how to control a complex robot subsystem and apply the principles in this paper to their own robots. However, these principles are not exclusive to FRC and can be applied in many other places in the real world.

1.4 What is a double pink arm with a wrist?

The double pink arm with a wrist is a spin on the original pink arm design, created by team 2383, the Ninjineers. It comprises of a single jointed arm, with a telescoping extension, and a wrist at the end. This allows for three degrees of freedom to score cones and cubes. Two neo's at the base control the pivot, two neo 550's control the telescoping extension arms, and one neo 550 controls the wrist at the very top.

2 Kinematics

2.1 Forward Kinematics

The three axes of the arm, will be denoted as following.

$$\theta_m, \theta_{wrel}, l \quad (1)$$

Where θ_m is the angle of the pivot joint, θ_{wrel} is the angle of the wrist joint relative to the arm itself, and l is the length of the arm. These three axes can be mathematically converted into an x, y, and θ_{wabs} coordinate system, where θ_{wabs} is the absolute angle of the wrist joint with regards to the x axis. The x and y values are the coordinates that the tip of the arm fall on. The conversion is as follows:

$$x = l \cos(\theta_m) \quad (2)$$

$$y = l \sin(\theta_m) \quad (3)$$

$$\theta_{wabs} = \theta_m + \theta_{wrel} \quad (4)$$

2.2 Inverse Kinematics

When controlling a mechanism like the pink arm, it is often more convenient to control the mechanism in terms of the x, y, and θ_{wabs} coordinates, rather than the θ_m , θ_{wrel} , and l coordinates. This is because the x, y, and θ_{wabs} coordinates are more intuitive to control, and are easier to visualize. It is much easier to measure values for the arm to reach and input them into the system rather than calculate the conversions by hand for every preset. However, the x, y, and θ_{wabs} coordinates are not directly related to the θ_m , θ_{wrel} , and l coordinates. This is where inverse kinematics comes in. Inverse kinematics is the process of converting the x, y, and θ_{wabs} coordinates into the θ_m , θ_{wrel} , and l coordinates. The inverse kinematics for the pink arm are as follows:

$$\theta_m = \arctan\left(\frac{y}{x}\right) \quad (5)$$

$$l = \sqrt{x^2 + y^2} \quad (6)$$

$$\theta_{wrel} = \theta_m - \theta_{wabs} \quad (7)$$

3 Feedforward Control

3.1 Introduction

Feedforward control provides a more accurate way to model the movement of a motor. It is a formula that models the dynamics of the system in question, with feedforward constants that are specific to each subsystem.

3.2 Kv and Ka

$$V = K_v \dot{x} + K_a \ddot{x} \quad (8)$$

Where V is the voltage, \dot{x} is the velocity, \ddot{x} is the acceleration, K_v is the velocity gain, and K_a is the acceleration gain. K_v determines the voltage required to keep the motor at a constant velocity, and K_a determines the voltage required to induce an acceleration on the motor. K_v and K_a are constants that are specific to each motor, and can be calculated by hand or by using software like System Identification.

3.3 Ks

In a frictionless world, the formula above works perfectly. However, in the real world, at very small velocities, the motor will not be able to overcome the static friction of the system. This is where K_s comes in. K_s is the static friction gain, and is the voltage required to overcome the static friction of the system. The updated motor feedforward equation with K_s is as following:

$$V = K_v \cdot \dot{x} + K_a \cdot \ddot{x} + K_s \cdot \text{sgn}(\dot{x}) \quad (9)$$

K_s is proportional to the velocity of the motor, and is multiplied by the signum of the velocity. This means that if the velocity is positive, K_s will be positive, and if the velocity is negative, K_s will be negative.

3.4 Kg

For the telescoping extension of the arm, the effect of gravity is negligible. However, for the pivot and wrist joint, gravity applies a significant force on the system, causing it to fall down if unmodeled. This can lead to a system that is inaccurate and hard to control. To combat this, a gain, K_g , is added to the feedforward equation. K_g is the gravity gain, and is the voltage required

to overcome the force of gravity. The motor feedforward equation for the pivot with Kg is as following:

$$V = K_v \cdot \dot{x} + K_a \cdot \ddot{x} + K_s \cdot \text{sgn}(\dot{x}) + K_g \cdot \sin(\theta_m) \quad (10)$$

The angle of the pivot is defined as 0 when it is vertically pointed up. As the angle of the pivot reaches 90 or 270 degrees, the effect of gravity is highest, which leads to the sine function outputting 1 or -1, leading to the most aggressive voltage compensation to counteract gravity. The formula for the wrist is very similar, but an important thing to note is that the angle of concern is the angle of the wrist with respect to the x axis, as gravity is applied in the y direction. A gravity feedforward that takes in the angle of the wrist relative to the arm will be inaccurate when the pivot is at any angle other than 0. The motor feedforward equation for the wrist with Kg is as following:

$$V = K_v \cdot \dot{x} + K_a \cdot \ddot{x} + K_s \cdot \text{sgn}(\dot{x}) + K_g \cdot \cos(\theta_{wabs}) \quad (11)$$

There is a cosine here because the angle of the wrist is with respect to the x axis. As the angle approaches 0 or 180, the cosine approaches 1 or negative 1, leading to the most aggressive voltage compensation to counteract gravity.

3.5 Conclusion

To conclude, the motor feedforward equations for the pivot, extension, and wrist are as following:

$$V_m = K_v \cdot \dot{x}_m + K_a \cdot \ddot{x}_m + K_s \cdot \text{sgn}(\dot{x}_m) + K_g \cdot \sin(\theta_m) \quad (12)$$

$$V_l = K_v \cdot \dot{x}_l + K_a \cdot \ddot{x}_l + K_s \cdot \text{sgn}(\dot{x}_l) \quad (13)$$

$$V_w = K_v \cdot \dot{x}_w + K_a \cdot \ddot{x}_w + K_s \cdot \text{sgn}(\dot{x}_w) + K_g \cdot \cos(\theta_{wabs}) \quad (14)$$

4 PID Control

4.1 Introduction

Although feedforward control is very accurate for modeled dynamics, it cannot account for all scenarios during an FRC match. When an unmodeled force hits the arm, the arm must be able to hit the setpoint despite the disturbance. This is where PID control comes in. As opposed to feedforward control, which does not use encoder values to dictate voltage inputs, PID control is a feedback control system that uses a proportional, integral, and derivative term to calculate voltage inputs using encoder measurements. The PID control equation is as following:

$$V = K_p \cdot e + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de}{dt} \quad (15)$$

Where V is the voltage to input into the motor, e is the error, or the difference between the set position and the actual position, K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, and t is the time. τ is simply a dummy variable for the integral.

4.2 Proportional Gain

The proportional term in the PID is meant to drive the positional error down to 0. To do this, the P term is proportional to the error. The higher the error, the larger the voltage, and this springs the motor position back to its setpoint. Conversely, the smaller the error, the lower the effort voltage. For some systems, this is all that is needed, and it is called a P loop, and looks like this:

$$V = K_p \cdot e \quad (16)$$

The K_p term can be tuned by hand, by running the motor and viewing graphs of the setpoint and actual position.

4.3 Derivative Gain

The derivative term works exactly like the proportional term. However, instead of being proportional to the error, it is proportional to the rate of change of the error. This means that the derivative term is proportional to the velocity of the motor. The higher the velocity error, the higher the control effort, just like proportional control. This term brings the velocity error down

to 0. Sometimes, this term is coupled with the proportional term, and is called a PD loop, and looks like this:

$$V = K_p \cdot e + K_d \cdot \frac{de}{dt} \quad (17)$$

The derivative term can be tuned just like the proportional term, by running the motor and viewing graphs of the setpoint and actual position.

4.4 Integral Gain

Although PD loops are very effective, these systems sometimes have a steady state error, or a constant error that is not 0. This is because the proportional and derivative terms are not enough to drive the error to 0. This is where the integral term comes in. The integral term is proportional to the area under the error curve, meaning that it gets more aggressive as time passes during steady state error. This is useful because during steady state error, the velocity is 0, and the proportional term is not enough to drive the error to 0, leading to no control effort during a period of error. The integral term is coupled with a P and a D term, completing the PID loop:

$$V = K_p \cdot e + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de}{dt} \quad (18)$$

However, designers should refrain from using the integral term, because feedforward control can usually avoid steady state error. The arm does not use the integral term in its PID.

5 Feedforward and PID Control

5.1 Putting it all together

In total, the entire control system for the arm is as following:

$$V_m = K_v \cdot \dot{x}_m + K_a \cdot \ddot{x}_m + K_s \cdot \text{sgn}(\dot{x}_m) + K_g \cdot \sin(\theta_m) + K_p \cdot e_m + K_d \cdot \frac{de_m}{dt} \quad (19)$$

$$V_l = K_v \cdot \dot{x}_l + K_a \cdot \ddot{x}_l + K_s \cdot \text{sgn}(\dot{x}_l) + K_p \cdot e_l + K_d \cdot \frac{de_l}{dt} \quad (20)$$

$$V_w = K_v \cdot \dot{x}_w + K_a \cdot \ddot{x}_w + K_s \cdot \text{sgn}(\dot{x}_w) + K_g \cdot \cos(\theta_{wabs}) + K_p \cdot e_w + K_d \cdot \frac{de_w}{dt} \quad (21)$$

6 Motion Profiling

7 Presets