Contents

Real Time Robot Motion Control

Conventional industrial robots are programmed to operate independently of human interaction in batch operations. In batch operations an industrial robot executes a fixed prewritten program to perform some operation such as, machining, welding and painting. The batch mode is similar to what (Computerized Numerical Control) CNC machines do to make parts using "G-code" as a prewritten program.

Real time motion control enables a robot to be trained to perform operations such as painting the exterior of a house or performing surgeries without relying upon prewritten programs. The user controls a virtual robot on the computer screen using a pointer in computer space (such as a mouse) or physical space (such as a 3D laser distance meter). The actual robot follows the virtual robot's operations.
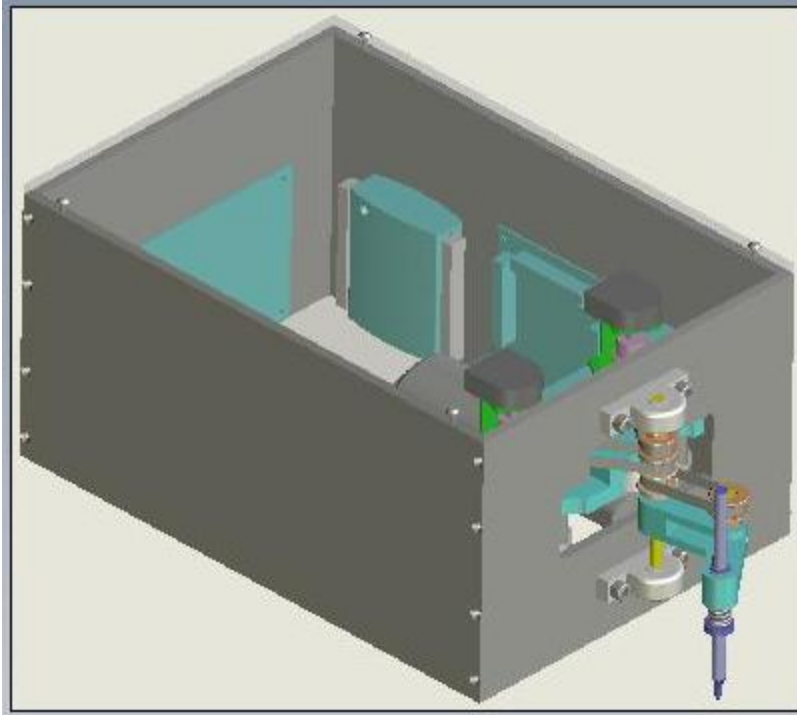
1

Figure 1 Flexbot Real-Time Motion Control System

Flexbot (Figure 1) is a sub-scale experimental real time robot motion control system under development. Flexbot introduces a "virtual reality" motion control technique. The user interacts with a "virtual" robot on the computer screen: 1) a virtual robot trains the user to maneuver the robot geometry in various modes 2) the virtual robot predicts how the actual robot will operate before physical motion occurs. In the appropriate software mode, the actual robot follows the virtual robot motions from the computer memory. Flexbot is comprised of computer software, sub-scale robot hardware, and servo drive electronics. The virtual robot motion (i.e. motion that is expected) and the predicted robot motion (the hardware motion in physical space) are compared on the screen.

Flexbot's software can be used in the classroom to train students in control theory (such as proportional, integral, and derivative (PID) gain settings) and to beta test real time robot motion control methodologies under development. Both activities help instructors to teach the engineering design process to students.

The Lego Mindstorms education products and MATLAB hardware in the loop (HIL)-Simulink development systems offer similar, but incomplete tools to implement the real time robot motion control system.
Demo software for the real time robot motion control software that is discussed in this paper can be found at this link:
http://www.laserpositioningsystem.com/Virtual_Robot_Demo/ArmCmdSim.zip

A description of how to manipulate the "virtual robot's geometrical positions follows, after which, a description of the how to control "virtual robot" hardware with PID controls settings is presented.

How to Manipulate the Virtual Robot Geometrical Positions

The Flexbot is a sub-scale prototype jointed-arm PC controlled robot.. Each of the Flexbot's drive motors are stationary to reduce the inertia of the arm's moving parts. The arm has two degrees of freedom in movement. The Flexbot is designed to demonstrate a virtual reality motion control technique under development. A mouse is used to control a virtual robot on a PC's computer screen. The virtual robot's position in the computer's memory is used to control the motions of the Flexbot hardware. The user points to a desired location on the screen. The virtual robot follows. In the appropriate software mode, the real sub-scale robot hardware follows the virtual robot by converting the digital information to analog commands. The Figure 2 is a block diagram of Flexbot's software and hardware interfaces:
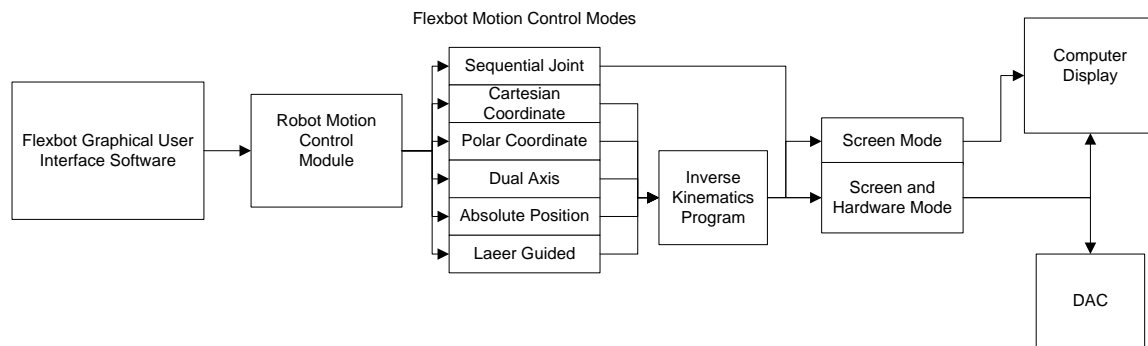

Figure 2 Flexbot Software and Hardware Interfaces

2D Real-Time Motion Control Software

Flexbot Software Operation Procedures (Screen Mode)

The Flexbot's graphical user interface (GUI) software has five modes of motion control:
1. Sequential Joint
2. Cartesian coordinates
3. Polar coordinates
4. Dual Axis Control
5. Absolute Position

Select the particular motion mode from the pull down menu in the upper left hand corner of the window. The Trajectory Mode dialog box on the upper right hand side of the window displays the selected Trajectory Mode. The default Trajectory Mode is "Sequential Joint Motion Control". See Figure 3 below
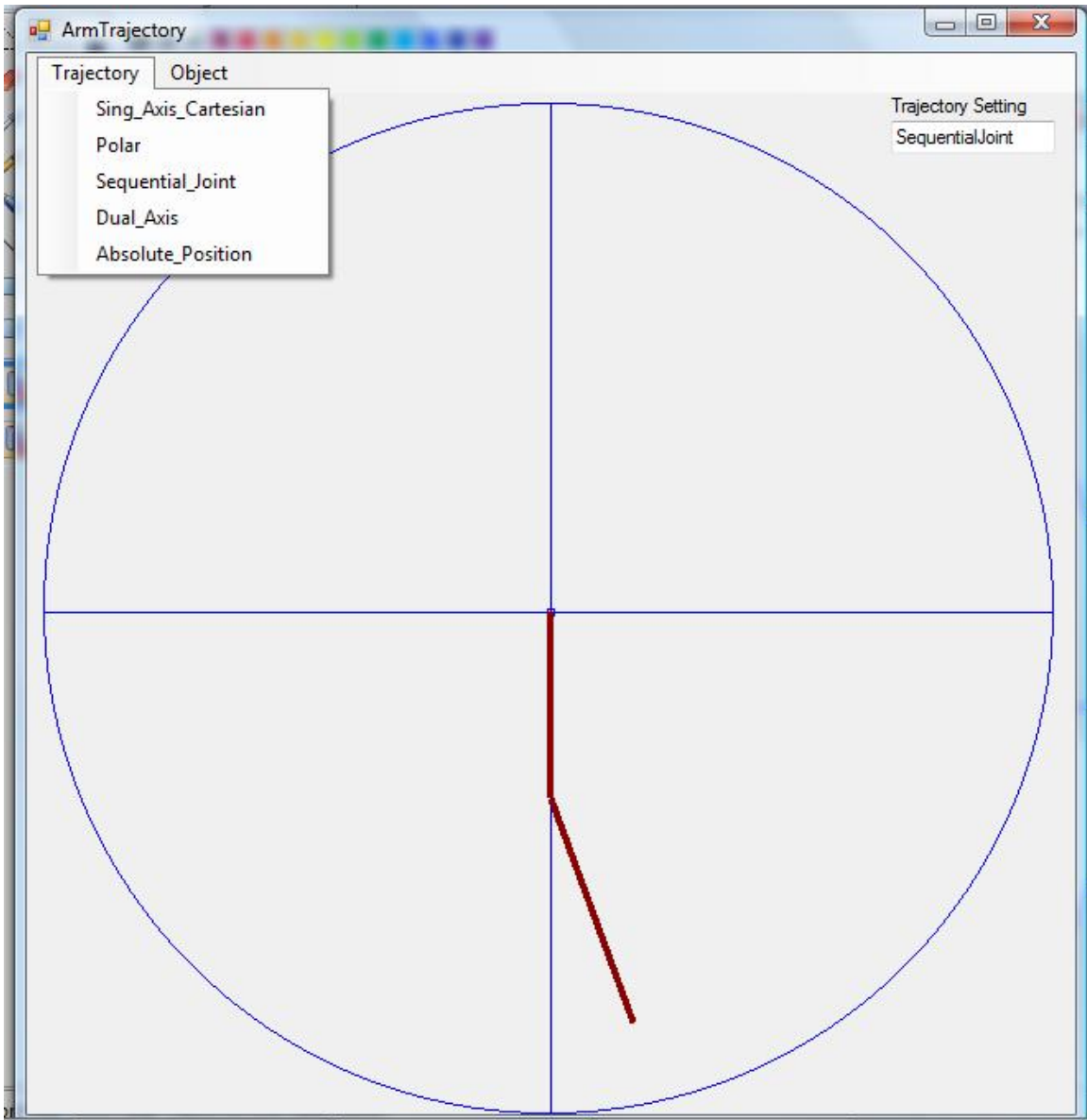
3

Figure 3 Screenshot of Flexbot's Graphical User Interface with Trajectory Modes Shown

Procedures for Each Control Mode:

1. **Sequential Joint Motion Control** (each joint is moved in sequence)
    a. Humerus or Shoulder Joint
        1. With the mouse cursor placed anywhere within the window, press and hold **left** mouse button. Move the mouse in the horizontal direction (left or right)
    b. Radius or Elbow Joint
        1. With the mouse cursor placed anywhere within the window, press and hold the **right** mouse button. Move the mouse in the horizontal direction (left or right)
2. **Cartesian Coordinate Motion Control**
    a. X-axis motion
        1. With the mouse cursor placed anywhere within the window, press and hold **left** mouse button. Move the mouse in the horizontal direction (left or right)

    b. Y-Axis motion
        1. With the mouse cursor placed anywhere within the window, press and hold **right** mouse button. Move the mouse in the **vertical** direction (up or down)

3. **Polar Coordinate Motion Control**
    a. Angular motion
        1. With the mouse cursor placed anywhere within the window, press and hold **left** mouse button. Move the mouse in the horizontal direction (left or right)
    b. Radial motion
        1. With the mouse cursor placed anywhere within the window, press and hold **right** mouse button. Move the mouse in the horizontal direction (left or right)

4. **Dual Axis Motion Control**
    a. With mouse cursor button placed anywhere on the screen, press and hold left mouse button. Move mouse cursor up, down, left, and right

5. **Absolute Position Motion Control**
    a. With mouse cursor button placed anywhere on the screen, press and hold left mouse button. Move mouse cursor up, down, left, and right. End point of virtual robot sticks to mouse cursor.

Real Time Robot Control System Software Description

3D Real Time Robot Control System Software

A YouTube video gives an overview of 3D Real-time operations at the following link:
http://youtu.be/gyHMvCit-Ng

Demo software of the 3D real-time robot control system for a typical six-degree of freedom robot can be found at this link:

http://laserpositioningsystem.com/Virtual_Robot_Demo/Self_Loading_6DOFRobot_Software.zip

Video instructions of how to use the 3D virtual robot software can be found at this link:
http://www.laserpositioningsystem.com/Virtual_Robot_Demo/6DOF_Demo3.html

The 3D "virtual robot" video shows a virtual representation of a three dimensional Laser Positioning System (LPS) can be found at these links:
http://laserpositioningsystem.com/images/

http://laserpositioningsystem.com/wp-content/uploads/2014/12/How-to-use-the-Laser-Positioning-System.pdf

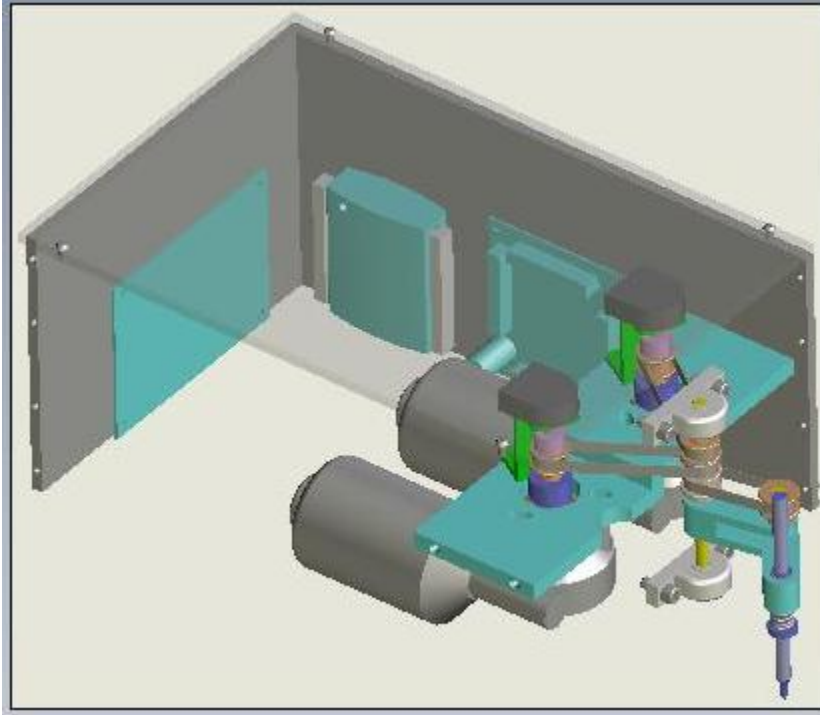## How to Control the Virtual Robot's Hardware (2D mode)

Figure 4 Flexbot Angular Position and Velocity Feedback Electronics

The robot real time hardware includes a controller, controller optimization parameter adjustments, a servo amplifier, motor electrical and mechanical time constants, motor angular position and angular velocity feedback, motor gear box. Figure 4 (above) shows the Flexbox angular position and velocity feedback electronics. The Flexbot instrumentation in Figure 4 includes high resolution encoders with quadrature decoders to keep track of angular positions. The quadrature decoders are part of the Encoder Counter Unit (ECU). The electronics include frequency to voltage converters that are integrated into the Encoder Tachometer Unit (ETU) to detect the motor encoder's shaft speed. Figure 5 (below) shows a block diagram of the real time robot motion control system.

Arm Movement Mechanism
The following paragraph is brief description of how the Flexbot robot arm moves. The Flexbot has two primary structural members:1) the humerus 2) and the radius. The humerus attaches to the shoulder joint (the main shaft mounted on the pillow blocks). The radius attaches at the elbow joint (the outer most pulley that is mounted at the opposite end of the humerus). There are three belts that are mounted to the main shaft. The top most belt rotates the humerus member. The bottom two belts rotate the radius member using an idler pulley on the main shaft as an intermediate step.

7

Development Status

Controlling the Flexbot hardware is the third step in using the real time robot control system. A new version of the motor speed feedback loop is now being implemented in experiments.
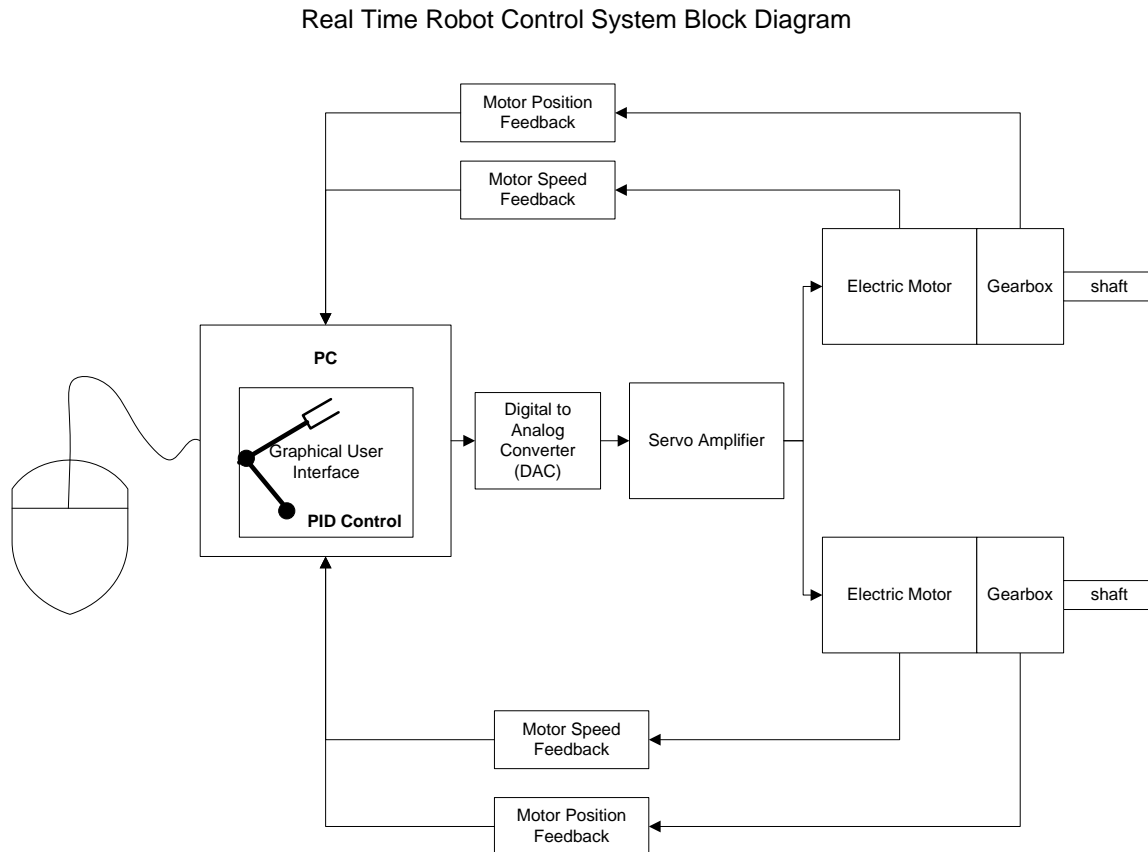
Real Time Robot Control System Block Diagram

Figure 5 Real Time Robot Motion Control System Block Diagram

Using Flexbot's Real Time Motion Control Software to Control its Hardware

To operate both the virtual robot and the hardware at the same time, choose the "Object" tab at the top of the window. Use your mouse to select the "Screen and Hardware" option in the pull down menu. See Figure 6 below.
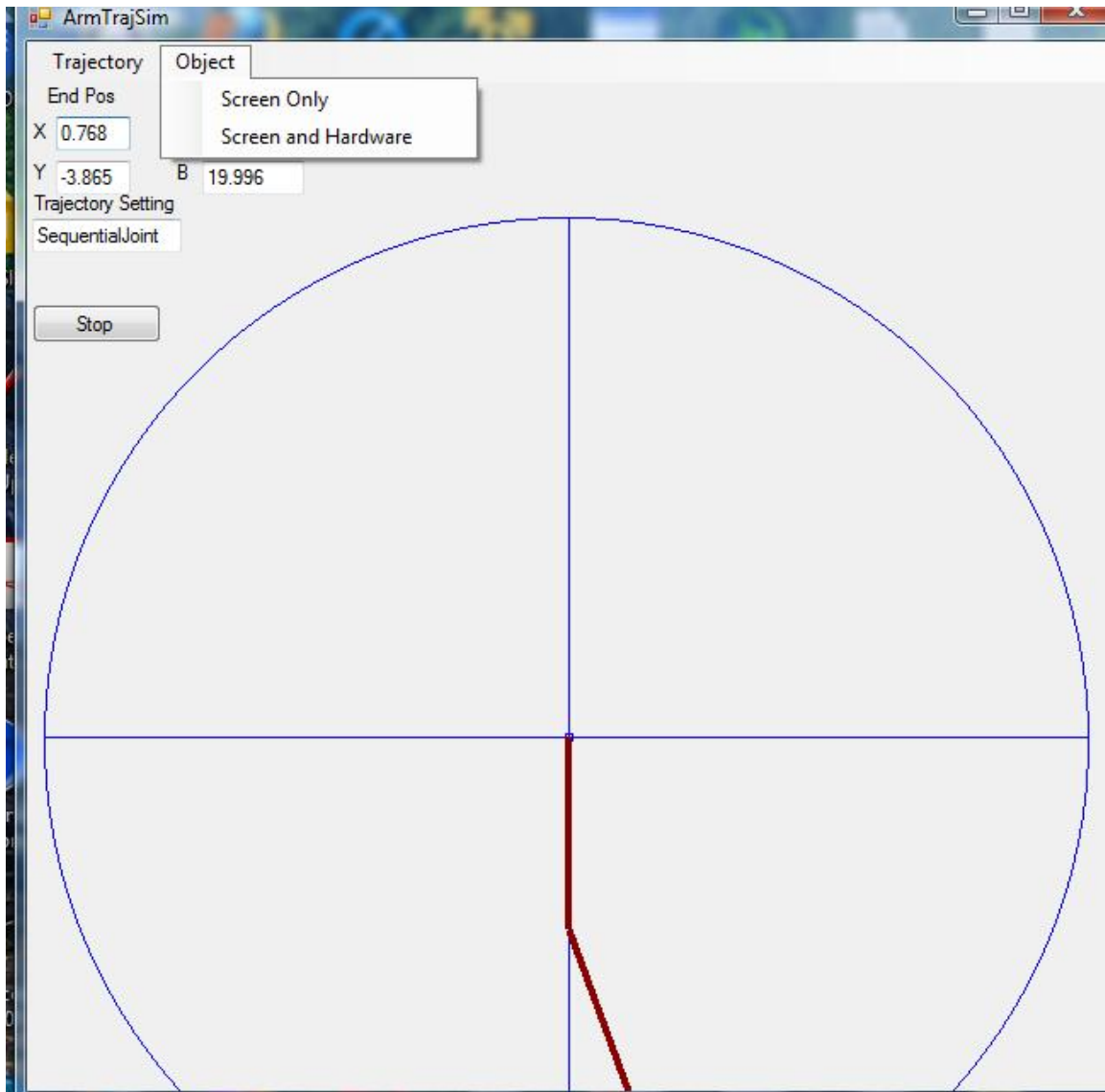
8

Figure 6 Making both the Screen and Hardware Move

More text boxes appear in the window to assist in tuning the hardware for the best response to a virtual robot position change. See Figure 5 below. The upper row of parameters is for the humerus member of the robot's arm. The lower row of parameters is for the radius member (forearm) of the robot's arm. The dark gray image is a representation of the location of the robot hardware. The "Angle Error" is an important measure of the hardware's (i.e. dark gray image's) ability to follow the lead of the red virtual (screen) robot.

The last three columns at the top of the window are proportional, integral, and derivative (PID) adjustment factors to optimize the hardware response to the red robot motions. For

9

example, the default PID factors are set at values that give the hardware a rapid response to the red virtual robot.

If you adjust the proportional gain factor, "Kpro", to "5" instead of the default "1000" value, the (dark gray) virtual robot hardware will respond very slowly by lagging behind the (red) virtual (screen) robot.

The integral gain, "Kint" factor is best set at a value near 0.0000001. Values appreciably above this will cause the hardware to become unstable and unable to follow the lead of the red virtual robot.

 Changing the Derivative gain factor," Kdrv" to 0 will cause the hardware to oscillate or overshoot the red virtual robot target position.
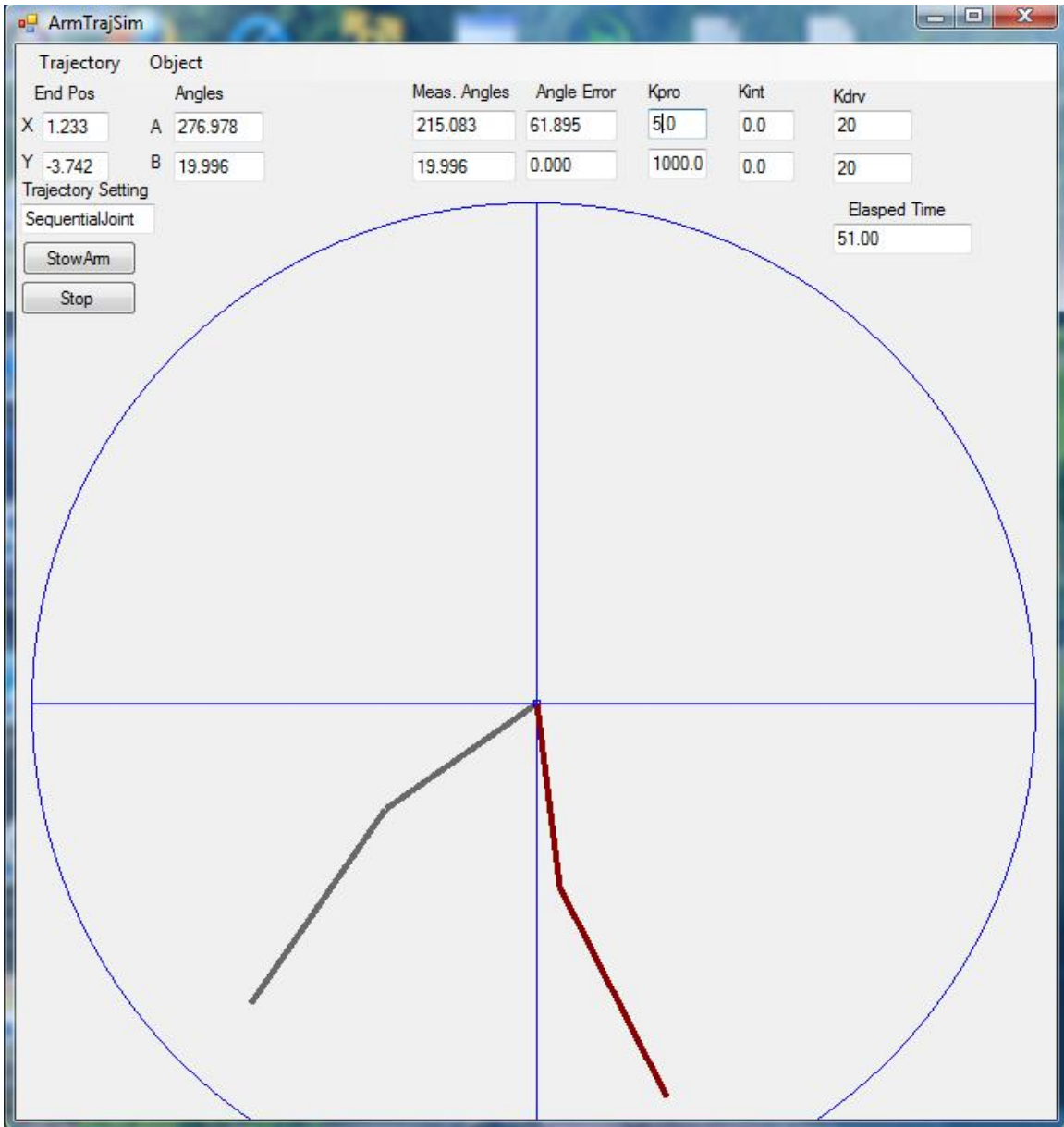
Figure 7, Virtual (red) and Simulated Physical Hardware (gray) Diagrem

Real Time Robot Control System Software Description
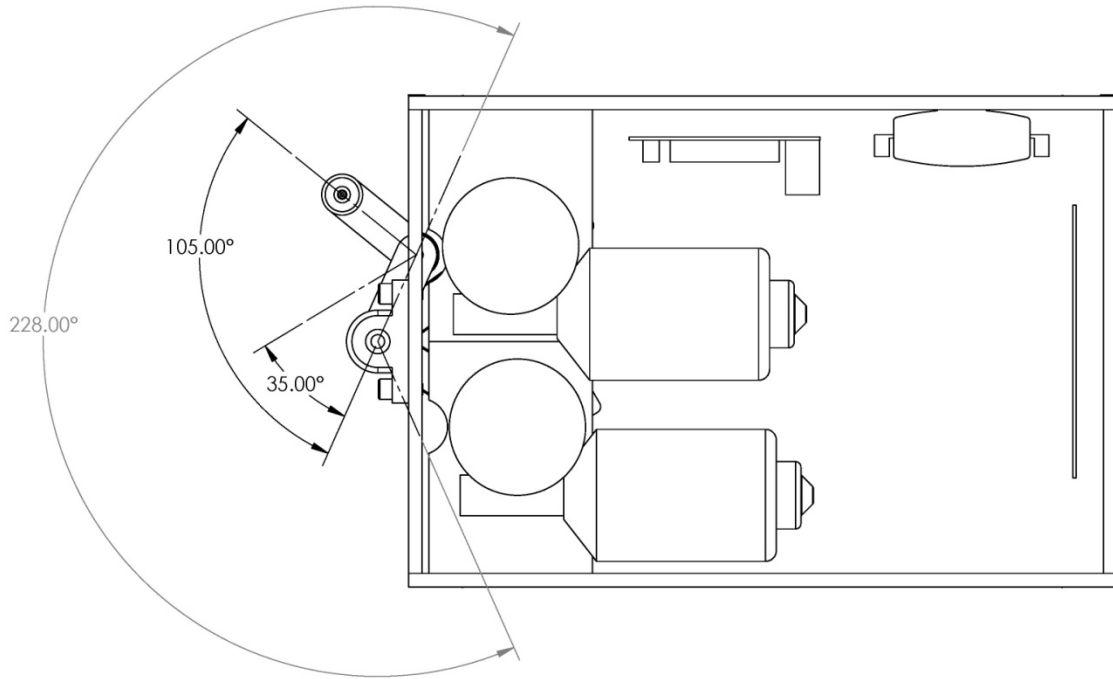
Flexbot's Mechanical Limits of Rotation

Figure 8 Flexbot's Radius and Humerus Mechanical Limits of Rotation
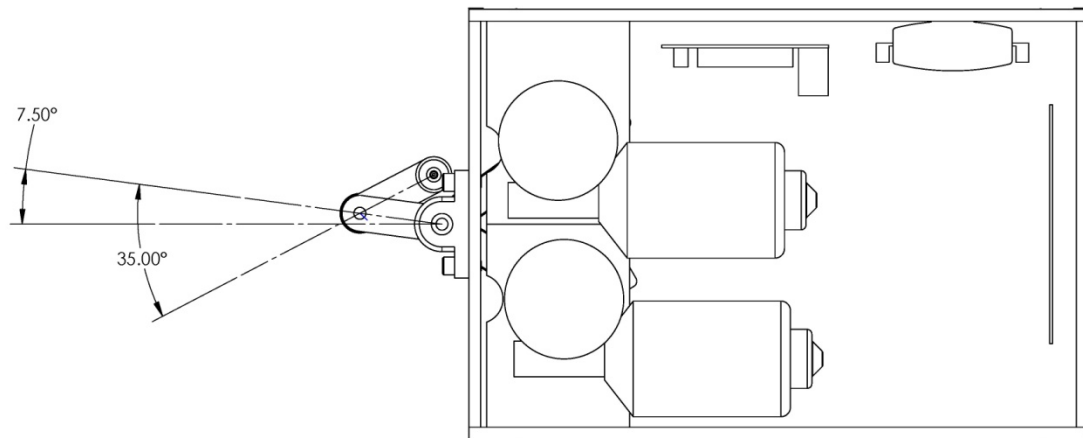

Figure 9 Flexbot's Radius Mechanical Limit of Rotation

The concepts introduced in this version of a real time robot control system are protected by US Patent No. 9,044,857

12