Architectural Robotics
Arch 879 / ECE 893
Spring 2009
Dr. Keith Green & Dr. Ian Walker
Intelligent Materials and Systems for Architecture (IMSA)
www.imsa-research.org

Project 3 – Supportive Environmental Technology for an Aging in Place Setting
J. Manganelli & Apoorva Kapadia

Title: ReLiS – Responsive Lighting and Screen Mount

Youtube URL: http://www.youtube.com/watch?v=KEGY0PPdznI

Abstract: ReLis is an intelligent home environment designed for heavy-use appliances to anticipate the needs of occupants. A redundant robot manipulator actively tracks occupant position and orients the media center display mounted on the end-effector. The lighting system adjusts lamp intensity, alleviating the need for the occupant to travel to fixed switching systems. The model presents a prototypical single occupant one-bedroom unit with a combined living room/kitchennette. The tracking algorithm adjusts display orientation and lighting intensity based on the pre-defined zone of the occupant location, allowing for minimal robot travel, intrusion, and obstacle avoidance, along with smart electrical power consumption.

Scenario: Miss Jane wakes in the morning, feels around for her glasses, and pulls her walker up to the bedside. A media center display mounted on a redundant manipulator notices her get out of bed and shifts to face her. Miss Jane proceeds to the kitchenette to make breakfast. Sensors mounted on the display passively track her movement and awaits her gaze. When she turns and faces it, it recognizes that she is looking directly at it and waits for instruction. Simple standard hand gesturing allows her to turn the display on or off, change the channel, adjust the volume and access her desktop or the web. As Miss Jane progresses through her daily routine, the system continues to passively track her position and await her gaze and instruction.

The system uses a combination of ultrasonic and PIR sensors to locate Miss Jane and determine her gaze. This allows her to interact with the media center from anywhere in the apartment with a single large display. Since Miss Jane has a fairly set daily routine, the robotic manipulator uses an intelligent algorithm to anticipate expected behavior. This allows the system, after just a couple of days, to preposition itself based upon her anticipated routine. In case Miss Jane is out or has fallen asleep, it knows where it should be when she returns or wakes up. In addition, the lights within the room are keyed off of the same input data and adjust automatically as she moves about her day.

While this scenario explores the use of this technology in the service of manipulating a media center display, our interest is in the capacity of such a system to be deployed for the intelligent manipulation of many essential appliances and devices. Other possibilities include a walker or smart nightstand that can recognize gaze and approach upon command in the event that the person is unable to approach the appliance.

Hardware:

- 3 servomotors
- 2 arduino boards

- 2 PIR sensors
- 3 LEDs
- 1 3-link redundant manipulator made of 26 ga. Sheet metal
- 6 6/32x2” screws
- 16 6/32 washers
- 4 1” nylon washers
- 4 1/2” O.D. Nylon spacers
- 1 5/16” steel rod
- 1 screen proxy

Code:

```
// ReLiS Motors 1 and 2 positioning code

// This is code that was used on two Arduinos.
// Arduino 1 controlled the Base and Middle Links.
// Arduino 2 controlled the End Effector and read from the PIR Sensors.

// While code here includes myServo3, it is not included in the set up
// as one motor shield can only power 2 Servo motors.
// Thus the declaration for myServo3 is present in the code for Arduino 2
// represents the only difference between the two programs.
// Because of this, only this code is included in our submission.

#include <Servo.h>

Servo myServo1; // Base Link
Servo myServo2; // Middle Link
Servo myServo3; // End Effector

int pos1; // Position of Base Link
int pos2; // Position of Middle Link
int pos3; // Position of End Effector

void setup()
{
  myServo1.attach(10);  // Base Link
  myServo2.attach(9);  // Middle Link

  pos1 = 0; // initial position of base link
  pos2 = 180; // initial position of middle link
  pos3 = 0; // initial position of end effector

  Serial.begin(9600);

}

void loop()
{
```

```
// Initial configuration
myServo1.write(0);
myServo2.write(180);
myServo3.write(0);

delay(3000);

// Zone 1 - Couch
// myServo1.write(0);
// myServo2.write(60);
// myServo3.write(90);

while(pos2 >=60)
{
  pos2 = pos2 - 1;
  myServo2.write(pos2);
  delay(100);
}
while(pos3 <= 90)
{
  pos3 = pos3 + 1;
  myServo3.write(pos3);
  delay(100);
}

delay(3000);

// Zone 2 - Kitchen
// myServo1.write(45);
// myServo2.write(135);
// myServo3.write(45);

while(pos3 >= 45)
{
  pos3 = pos3 - 1;
  myServo3.write(pos3);
  delay(100);
}

while(pos2 <= 135)
{
  pos2 = pos2 + 1;
  myServo2.write(pos2);
  delay(100);
}

while(pos1 <= 45)
{
  pos1 = pos1 + 1;
```

```
  myServo1.write(pos1);
  delay(100);
}

delay(3000);

// Zone 3 - Recliner/Workstation
// myServo1.write(90);
// myServo2.write(90);
// myServo3.write(30);

while(pos3 >= 30)
{
  pos3 = pos3 - 1;
  myServo3.write(pos3);
  delay(100);
}
while(pos2 >= 90)
{
  pos2 = pos2 - 1;
  myServo2.write(pos2);
  delay(100);
}
while(pos1 <= 90)
{
  pos1 = pos1 + 1;
  myServo1.write(pos1);
  delay(100);
}

delay(3000);

 // Zone 4 - Bedroom
// myServo1.write(90);
// myServo2.write(90);
// myServo3.write(90);

while(pos3 <= 90)
{
  pos3 = pos3 + 1;
  myServo3.write(pos3);
  delay(100);
}

delay(3000);

while(1)
{
}
```

}

Lessons Learned/Future Work:

Arm Stiffness -- The first arm design was undersized and unstable when supporting the load. In particular, we were using bent aluminum tubing for structure and once bent, its strength was significantly compromised. The redesign uses folded sheet steel and is both light and rigid. Nonetheless, even this required the addition of stiffener plates to deal with moment when the arm is fully extended under load.

Motor Slippage -- Motor slippage hindered our ability to accurately and consistently control location and orientation of the end-effector. The problem is mounting the links to the nylon gearing extending out of the top of the servo. (The servo motor itself is strong enough and the positioning accurate enough to position as needed.) However, control diminishes as the nylon gearing wears. We solved this problem by using set screws in the link posts that mount over the nylon gear. Now, in addition to the posts being threaded into the nylon gear from the top, they squeeze the nylon gear on its sides and slippage is eliminated.

PIR Sensor Suitability -- We worked with PIR sensors because of their ease of use and availability. Our preferred sensors would be ultrasonic but these were not within the budget. Using the PIR sensors presented technical challenges. They are very sensitive. But calibrating them with the motors proved to be very time-consuming and returned limited results so we did not progress past the testing phase with the PIR sensors. If we were to take this project further, we would invest in the ultrasonic sensors as we think that they are the better overall choice.

Coordinating Development of the Manipulator and the Software -- As noted above, we encountered several technical issues, such as the slippage issue, for which we had to redesign the arm on the fly before we could move forward with calibrating the code and the device functionality. Thus our typical workflow involved getting the arm to a point, starting to test the code and then tweaking the arm or the code as required. Through this iterative process, we were able to refine and control manipulator functionality. So far, we have rebuilt the arm about four times and reworked the code several times as well.

Building a Controllable Redundant 3-link Manipulator Versus Developing a Novel Learning Algorithm for the Application -- Our initial intent was to focus on developing the learning algorithm. In order to do so we first needed a functioning and controllable manipulator. Designing and building such a device proved to be much more involved than we initially estimated. As a result, our project has become the design, construction and programming of a simple 3-link redundant manipulator. If we were to do it over again, we would use an existing manipulator in order to focus on the novel coding and addition of sensors.