# ECE 479/579 Intelligent Robotics II, Winter 2012 Term

# Projects List

1. The mechanical/body design of MCECS\_BOT (hardware and software) **Omar Mohsin**
2. The head and neck of the MCECS\_BOT (mostly hardware)
3. The hand of MCECS\_BOT (hardware and software, mostly hardware)
4. Person recognition with Kinect (software)
5. Emotion and Gesture recognition with Kinect (software) **Hoang Nguyen**
6. Age/Gender recognition with Kinect (software) **Hamed Mirlohi**
7. Integrated dialog system based on vision and speech recognition/ text from keyboard for MCECS\_BOT with Kinect (software) **– Robert Fiszer**

Most of these ideas come from Mathias Sunardi. I will discuss my related ideas in class and they will be also in the book. However, this is your project and I expect creativity from you, what you find here is only the first step and the plan of action.

I will assign presentation lectures for each student. You will have to deliver these presentations during the whole time interval of the class, the last in the week of final which will be part of the official presentation to the Dean of Engineering.

## Project 1

## Mechanical/Body design of MCECS\_Bot

In this project you will be dealing with two major parts of the MCECS\_Bot:

1. The mobile base.
2. The robot 'body'.

The Mobile base is an off-the shelf scooter-like vehicle. Like a Segway, but less cool (unfortunately). But you can make it cooler - by adding a robot to it! You will have to modify/hack this mobile base so it can be controlled by the MCECS\_Bot system. Specifically, you will have to think of a way to control the speed and direction of this scooter remotely (mechanically and electronically).

The MCECS\_Bot body should have two main parts: lower and upper. The upper part provides mounts for the neck/head, two arms/shoulders, and a display/LCD screen (approx. 10" screen). It will be fixed in relative to the mobile base, but able to twist left/right (yaw). It is not necessary to be anthrophormically male/female anatomically correct, but should be somewhat 'organic' (think as less blocky).

The lower part of the body will be freely attached to the upper body and mobile base. The goal is to have the lower body moveable, to simulate dancing (e.g. at the hip). 'Freely attached' means it is connected with at least one degree of freedom. However, it is not necessary for the lower part to be actuated AT the DoFs on the upper body and mobile base. Some creative ways to move the lower body is desired. It is not necessary to have the lower body to be bipedal. creativity is encouraged!

Additionally, think about the following:

1. Where to put the controller board for the servos. You might need to create mounts for it.
2. Where to put the power supply.
3. How the controller board (maybe a motherboard) and power supply will be connected.
4. If you need to have a custom 'hub' to connect everything together.

We have some resources, tools, materials in the Robotics Lab. Possibly we can have help from experienced mechanical design engineers from PARTS and PSU Department of Mechanical Engineering.



Hips, knees, ankles are free to move while the upper body stays still, but .…

LCD/Touch screen to display face/facial expression

LCD/Touch screen to display information

… ‘Feet’ are fixed to the base

Upper Body is fixed relative to the base

(maybe at the ‘pelvis’ center)

Figure - Rough concept of the mcecs-bot (not shown: keyboard, power supply, kinect)

## Project 2

## Head and Neck of MCECS\_Bot

We will have two versions of the MCECS\_Bot head:

1. LCD Screen/Tablet. The screen will show a cartoon face of MCECS\_Bot that can display different facial expressions (pre-programmed). We envisioned it to be a touchscreen so users can interact with it and annoy the robot in the process. Think of the freedom of what you can have with graphics! The programming of the faces will be a separate project. Check out MIT's Kombusto (<http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/video-kombusto-mit-interactive-dragon-robot>) for inspiration/reference.
2. A puppet/physical head/face. It might be a more involved, mechanical project but not necessarily so. We suggest a simpler, puppet-like (i.e. Muppet-like) head instead of human-like with high degrees of freedom. More degrees of freedom requires more actuators (in this case, servos) which add weight to the head, which makes the neck design will be more difficult, and so on. Start with one degree of freedom for the mouth, and another for eyebrows - that's it. Check out Sparky robot (<http://markscheeff.com/engineering/engineering_projects/sparky/papers/WIRE2000_final.pdf>) for inspiration/reference. Remember - minimal actuation for maximal freedom.

I'm going to brainwash you here: Head gestures are more interesting than facial gestures. Trust me on this. SO, for MCECS\_Bot, we want you to focus on the degrees of freedom for the neck. Strive for having 3 degrees of freedom for the neck for tilt, yaw, and pitch. Ideally, we would like to have both versions of the head, so if you do:

1. The LCD Screen/Tablet - the 'head' will be just a tablet mount (e.g. iPad mount).
2. The puppet/physical head/face - well, ideally it would be nice to be able to swap heads. But you decide.



There must be a mount for the LCD/Touch screen here

DOF for Yaw

DOF for Pitch

DOF for Tilt

Figure - neck example

We have some resources, tools, materials in the Robotics Lab. Possibly we can have help from experienced mechanical design engineers from PARTS and PSU Department of Mechanical Engineering.

## Project 3

## The arm and hand of MCECS\_Bot

Again, think simple. For the hand, we recommend very simple mechanics - one degree of freedom (e.g. gripping) will do. However, if you can, try to make the shape of the hand a bit more organic instead of a simple claw. The arm, however, can be more interesting, especially the shoulder. An expert in biomechanics once told me, modeling the shoulder is one of the most difficult thing in the biomechanics field - a perfect model of it is like the Holy Grail in biomechanics. Similar to the neck/head, we encourage you to less focus on the anthropoid hand and digits (fingers) with millions of degrees of freedom. Instead, focus on the mechanics and articulation of the arm and wrist. Try to use as little (active) actuation as possible, and use ideas from tinker toys where one actuation can be used to move multiple parts at the same time. Yes, it will limit the amount of control you can do with the arm, but if you do it right, we can have an arm that can perform very convincing, expressive gestures.

Recommendations:

* + Focus on the shoulder and wrist: 3 DOF on the shoulder, 2 DOF on the wrist, 0 or 1 DOF on the elbow.
	+ Do one arm first, but keep a good record of its blueprints

… ‘Feet’ are fixed to the base

Maybe 2 DOF here, but at an angle

 

Simple hand (open palm), no digits

2 DOF wrist

FIGURE 3 – ARM & HAND EXAMPLE

We have some resources, tools, materials in the Robotics Lab. Possibly we can have help from experienced mechanical design engineers from PARTS and PSU Department of Mechanical Engineering. I will give you more advanced and detailed plans and ideas how to build various robot hands. Remember this hand is for gesticulation not manipulation, so manipulation can be very restricted. You can also add fingers with few degrees of freedom but just for gesticulation, not grasping.

## Project 4

## Person Recognition

This is a software-focused project, and your programming skills will be forged with this project (if it hasn't already). The deliverable of the project will be a C/C++ program that will take an image (i.e. frame) from the Kinect camera, process the image, and return a Boolean True or False indicating whether or not a person is present in the image, and if possible, who the person is. The Person Recognition is a feature that will be implemented in the MCECS\_Bot guide robot project as part of its human-robot interaction component. The presence or absence of a person will determine how the robot will behave (this will be a separate project, but talk to me if you're interested in it). One application we envisioned, is when the system recognizes the Dean, faculty, or staff, it will be able to introduce them to the other person who is a guest.

Some of the things you will be dealing with in this project:

* + installation of the OpenKinect, MRPT, and (maybe) OpenCV libraries
	+ C/C++ programming
	+ Face detection algorithm (see Viola-Jones paper)
	+ Face recognition algorithm (identifying who the face belongs to)
	+ Classifier

Resources/references for this project:

* <http://openkinect.org/wiki/Research_Material>
* <http://www.kinecthacks.com>
* <http://opencv.willowgarage.com/wiki/FaceRecognition>
* <http://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework>
* <http://research.microsoft.com/en-us/um/people/viola/Pubs/Detect/violaJones_IJCV.pdf>
* <http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CB4QFjAA&url=http%3A%2F%2Fwww.ece.pdx.edu%2FNewsCurrentEvents%2FThesis%2FDimitriyLabunsky_MS_Defense_11-6-9.pdf&ei=sigKT9TaJYKkiQLtkI3ICQ&usg=AFQjCNGpQ7IkMEd2Ty830k2sSor8xaMzGg> (Dimitry Labunsky MS Thesis)

## Project 5

## huahoang2112@gmail.com>:

## EMOTIONS AND Gesture Recognition

This is a software-focused project, and your programming skills will be forged with this project (if it hasn't already). The deliverable of this project will be a C/C++ program that will take a sequence of images of a person from the Kinect, analyze the sequence of images of facial expressions, body or arm poses, and identify and returns the type of emotional gesture that was performed by a person in the image sequence.

You have to read several documents about facial emotion recognition, study about Kinect and its application in movement recognition and 3-D shaping; there're an Kinect SDK from Microsoft used to develop application for Kinect that can be used with Microsoft Visual Studio. There are many other.

Recommendation:

* 1. First connect Kinect to Labunsky’s software
	2. Start with one or two facial expressions/gestures, but aim to have about at least 5 identifiable gestures in total. The ideal is seven, as in Labunsky’s thesis.
	3. Do not worry about subtle gestures (e.g. shoulder shrugs) - aim for easy, obvious body gestures like pointing or hand wave.
	4. You may need to create a database of gestures to be matched against. You can probably find some database available online.

Some of the things you will be dealing with in this project:

* + installation of the OpenKinect, MRPT, and (maybe) OpenCV libraries
	+ C/C++ programming
	+ Human body segmentation algorithm
	+ Classifier

Resources/references for this project:

* Labunsky’s thesis from PSU (Nguyen has it)
* Documentation of Kinect (students from 478)
* <http://openkinect.org/wiki/Research_Material>
* <http://www.kinecthacks.com>
* <http://channel9.msdn.com/coding4fun/kinect/Open-source-Kinect-gesture-recognition-project-Kinect-DTW>
* <http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

## Project 6

## Age/Gender Recognition

This is a software-focused project, and your programming skills will be forged with this project (if it hasn't already). The deliverable of this project will be a C/C++ program that will take an image of a person from Kinect, and classify the person in the image as a child or adult, a female or male.

Recommendation:

* + Prioritize on classifying the person's child or adult status first, then gender. Because … people tend to get sensitive if you misclassify them.
	+ This is unexplored territory for us. So if you are doing this, we applaud you and looking forward to your results.

Resources/references for this project:

* <http://openkinect.org/wiki/Research_Material>
* <http://www.kinecthacks.com>
* <http://www.youtube.com/watch?v=T-Pb_M3myAw>
* <http://www.svcl.ucsd.edu/~nikux/age/>
* <http://www.geekwire.com/2011/microsoft-idea-kinect-body-scans-estimate-age-automate-parental-controls>

## Project 7 (Robert FISZER)

## Integrated dialog system based on vision and speech recognition/ text from keyboard for MCECS\_BOT

This is a software-focused project, and your programming skills will be forged with this project (if it hasn't already). The deliverable of this project will be a C/C++ program which will take inputs from Person Recognition, Gesture Recognition, speech input, and/or text input, process all those input information to extract context and meaning of the current interaction, and determine and produce an appropriate response to the user in the form of a sentence (i.e. text which can be synthesized as speech), and/or head/neck, arm, and body gestures. You will base the dialog system on the Natural Language Processing program developed by Mr. Robert Fiszer. This system will be one of the core sub-system for the MCECS\_Bot - which means it is a very, very important component of the MCECS\_Bot.

This project has several sub-projects:

1. Speech recognition system
2. Text-to-speech system
3. Natural Language Processing
4. Integration of the sub-systems

Recommendations:

1. If you're working on the Speech recognition system:
	1. Use the CMU Sphinx or Pocket-Sphinx software.
	2. Train it with simple sentences or words "Me Tarzan, you Jane" kind - do not concern with everyday sentences in the very beginning.
	3. You can get away with creating a new 'language' which is a subset of English grammar
2. If you're working on the Text-to-speech system:
	1. Use the FreeTTS software
	2. Try to make it work under C/C++
	3. See if you can find ways to play with the voice modulation so it sounds a bit more 'natural' than 'robotic' (e.g. Siri)
3. If you're working on the Natural Language Processing:
	1. Robert Fiszer will be the lead in this project. Talk to him if you want to be a part of it.
	2. Assume the inputs from the other systems. E.g. the Person Recognition (Yes, No, perhaps accompanied by a person's name: John, Mary), Gesture Recognition (Greeting, point/direction), Speech and text (assume a set of sentences: "Hello, robot", "Take me to the Robotics Lab", "Where is the men's restroom?", "The sky is blue.")
	3. You may create a small program that would spit out those assumed inputs
4. This is for all of you involved in Project 7. As you develop your sub-projects, you should be thinking how to make it easy to integrate your part with the others.