

**1) robot hands to get data, how can we gather that?**

to collect the data. You only need your own hand. The kinect sensors is going to instrument. It can get all your joint angles.

Do we need to use it in the lab.

take the kinect actions with your own hand and then instrument it.

we want to give room, come up with some body movement sensors as a replacement for kinect.

if you want to come up with different body mounted sensors.

you have libraries.

university is working at 50% capacity. We have lots of time constraints.

body variable sensors, you have to construct it.

we expect a nice blend between software and hardware implementations.

Action dictionary.

expected to use off shelf AI algorithms, and then

**2) what is the interface between the Robotic hands and the ML algorithm**

coppelia, make it really easy.

**3) could you provide us with a baseline skeleton for the code of the project?**

paper on imitation learning.

In terms of code I was using the Glove, link to github.

rudimentary code, for kinect that performs

**4) resources for theory / inspiration**

**5) we need to set up a date for weekly meetings**

**6) computational resources to deal with Deep Learning Algorithms?**

---

embedded and network system group:

Vineet Gokhale expertise: tactile internet and wireless communications.

primary point of contact for this project

Ashu: sort of supervisor for this course.

Tactile internet, there is also Force, or tactile feedback that is sent.

Low latency demanding.

Typical audio and video communication, there is seconds of Latency.

You will be able to grab an object precisely only if the latency is in the order of 10-20 ms.

The current internet can not guarantee such latency, it's not designed.

Speed of light is a barrier and can not go beyond that.

How do we enable

4. object of this project.

The idea is that instead of transfer position and velocity.

Why not enable an edge intelligence, a server that is very close to the human operator, and a server that is close to the robotic operator.

These servers run some kind of intelligence in order to predict the signal and pass the information a bit before.

Building such intelligence requires one to learn how human operator moves his hand to perform an action.

Using this data. We can deploy this intelligence to introduce "negative latency".

Telly operator that plays ping pong.

Ofc if the robot is in front of you. Np. Else you have lots of problems

First, we call it offline learning. Where the robot is in close proximity to the glove and the arm that you will be using.

Microsoft kinect sensors that is going to instrument all your arm. All the joints are going to be .

Whenever you make any motion. You will save the angles.

So that it can learn from that action so that the next time you perform that actions you can mimic it.

We have 2 AI blocks, 1 at the end of the human, 1 at the robot.

Scenario, comes up.

The robot already knows what the human will do.

If the ball comes at a certain velocity with an incoming angle.

As soon as the human moves, giving some clue. The robot can predict what kind of shot the human is going to make.

The robot is going to categorize the action.

Classify the action to a low dimensional space.

Dictionary based AI.

Whenever you have a large high dimensional set of signals, and action is possible to parametrize them to a low dimensional space.

Radial Basis Function kernel to codify your entire possible set of functions.

Constrained to a set of context.

Any action corresponds to a low dimensional set of parameters.

The AI is going to mimic that action before it performs. Then correct itself, reparametrize itself, or maybe to a higher dimensional space.

The codified trajectory is .

Online learning. Codify trajectory has to be corrected.

You will have a Kinect to which you can measure robotic angles. Then the robotic hand can be replaced by a virtual platform. Simulated robotic arm that simulates the robotic movement.

Basic level of the student to complete. If you are able this well in time. You can construct the physical version.

- 1- Basic level: you must complete
- 2- Slightly advanced: from virtual environment to physical setup (any support, getting components it's okay)