Multimodal Human-Machine Interaction for Human Adaptive Manipulation

Technical Report: CES–507

Hossein Farid Ghassem Nia,

James A. R. Cannan,

Mohammadreza Asghari Oskoei,

and Huosheng Hu

July 2010

School of Computer Science & Electronic Engineering

University of Essex

Colchester CO4 3SQ, United Kingdom

Introduction

Robot Teleoperation, so-called Telerobotics, indicates manipulation of a robot at a distance. It has a wide range of application in surgery, aerospace, hazardous area, simulation and training. Two major components of Telerobotics are the sensing and control applications. Remote sensors provide the information as an input reference into the control unit, which directly manipulates the robot. The performance of the Telerobotics can be evaluated based on the accuracy (i.e. resolution), intuitiveness, and response time.

This project represents a robotic arm teleoperated through visual and EMG data provided in a remote location. The robotic arm imitates a human's arm action in displacing an object on the table using visual data provided by a fixed camera on top of the table, and EMG data collected from human arm's muscles. It is designed to teleoperate the robotic arm in a potentially hazardous area by anybody just after a short-term calibration process.

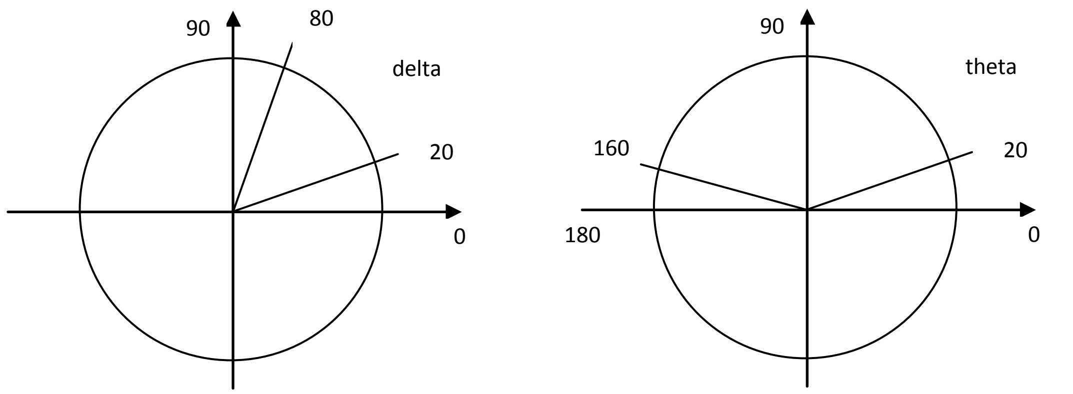
System Design

The aim of this project is to develop a setup in which a robotic arm imitates a human arm's action in real-time. The tele-operated robotic arm simulates human arm action in displacing an object on the table. It grabs, relocates and releases an object on the table. The robotic arm is featured by an advanced tele-control system that adopts biomedical signals as well as visual data. The setup is built up of two tables: robot and work tables. A robotic arm with two degrees-of-freedom (DOF) and a two-finger grabber is fixed over the robot table. The work table, likely placed in a safe and remote area, is featured with a fixed camera on top, which track an object motions in the work area (i.e. part of work table which is commonly covered by the camera). Detected motions by visual data processing provide information to control the robot arm. Meanwhile, data collected through an EMG electrode located on the user's arm is processed to discriminate hand status (i.e. open or close) and supply additional information to control the grabber connected to the robotic arm.

The robotic arm has two joints to perform two actions of yaw and pitch. The first joint fixed to the table produces yaw action by rotating the robot. It rotates the robot arm between 20 and 160 degrees, and generates the yaw angle (theta). The second joint connects the first joint into the grabbing device. It produces pitch angle (delta) varying between 20 and 80 degrees. The angles (theta and delta) are both measured in counter clock wise. The last part is a two-state grabbing device formed by two fingers. The grabber can be either closed (to grab) or open (to release). The initial state of the robotic arm is defined in yaw angle 90, pitch angle 60, and the grabber in the state of open (theta=90, delta=60, grabber=open). Actions defined for the robotic arm are as following:

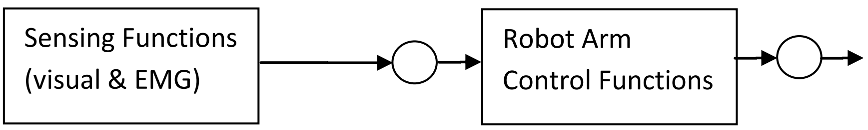
- Grab/Release: turns the grabber into closed/open state.

- Pitch: changes the robot arm delta angle.
- Yaw: changes the robot arm theta angle.



Left: view of work table made by side camera Right: view of work table made by top camera

The sensing functions are comprised of object detection/tracking applied to visual data and double threshold function applied to EMG signal. Visual data is used to localise and track the object in the working area, and feed the desired reference values for theta and delta angles. EMG signals are passed through a threshold based controller to discriminate the hand state and provide the desired states for the grabber.



Vision

As mentioned, vision and EMG are main sensors to control the robotic arm in this experiment. The vision system in this experiment, tracks an object in 2D space to detect an object position. A camera is installed at the top of the work area (a rectangular box 1x1m) at a height of 60cm. The camera is a Logitech 1.3 mega pixel webcam which is connected through a USB port to the host pc.

Image processing is done by using OPENCV library and functions of this library are used for object detection. To detect an object, different methods have been tested. At the beginning of this experiment, object detection was implemented by colour detection. But this method faced many problems in the development phase of project. Object detection using colour thresholding is not the best option. In different tests, most of the time the system can detect colour easily but in the case that light condition changes or the environment contains noise, performance decreases significantly. Another problem of object detection by using colour thresholding is the problem of finding correct and constant thresholds for different objects.

After conducting several experiments, it is found that colour thresholding is not a proper method for this project. Therefore it was decided that a template matching and a circle detection method be used. Template matching is a technique in computer vision for finding small parts of an image which match a template image. For this purpose, an orange colour ball was chosen as an object, and by taking the ball's image a template image was created. By using OPENCV functions for template matching, the program looks for templates in a sequence of images, with the expected outcome of object detection and tracking in a sequence of images.

To obtain better results, it was decided to use circle detection in addition to template matching. Circle detection is implemented by using the Hough circle function in OPENCV. The function finds circles in a gray-scale image using Hough transform. The idea of the Hough transform is that perpendiculars to edge points of a circle cross in the centre of the circle. Circle detection performed well but in the cases that the object is moving fast, it cannot detect it accurately.

Information yielded by both template matching and circle detection are fused together to produce robust and reliable data which is used in controlling the robotic arm. The proposed vision system is not sensitive to light condition and has an acceptable performance in a noisy environment.

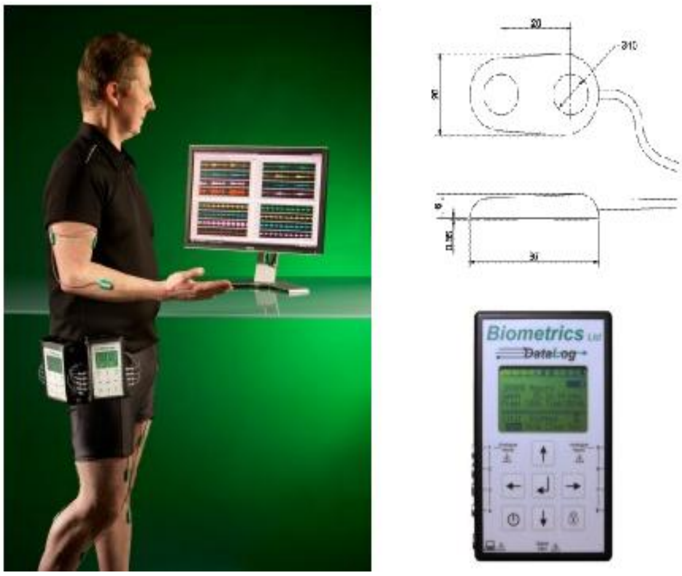
EMG

Electromyography (EMG) is a technique used for monitoring electrical signals generated by muscles. Generally these signals are only a few milliVolts in amplitude and so require to be amplified close to 1000 times. The EMG signal resides in the 0-500hz frequency band and tends to be quite noisy, which is why it is often filtered into the 15-450hz range.

The EMG hardware used in the project was the Biometrics DataLog system, which is a fully portable Bluetooth enabled data acquisition unit. It allows up to 8 simultaneous EMG channels to be monitored, and uses active differential electrodes with up to a 9Khz total sampling time. It monitors the same frequency mentioned above (15-450hz)

Once the EMG signal is acquired it is rectified and averaged to produce a stable output, simple thresh-holding techniques are then employed to activate the robotic arm sequence to grab or release an object. Currently only a single channel is required to detect the user's intentions. The simplicity of the system allows the electrode to be placed on the wrist, forearm, bicep or any muscle surface that produces enough signal amplitude to pass the threshold, this gives the user the freedom to position the electrode where ever the user finds most comfortable. The current design only uses the EMG signal as a switch, allowing anyone with muscle control to use the system whether they are physically disabled or not.

Future work might employ more electrodes with enhanced signal processing if further complex operations are required, this would improve on the fine control and stability of the system, and would also allow the possibility of a robotic hand being used for improved object manipulation.



www.biometricsltd.com

Robotic Arm

The robotic arm which is used in this project is the EduBot arm, it has five degrees of freedom and a gripper. The gripper on the end of the robot allows the robotic arm to pick up small light weight objects. The Edubot arm was chosen for its flexibility, light weight and fast response time.

www.robotica.co.uk

There are restrictions imposed on the robot. Firstly it can only move within a 180 degree semi-circle, and secondly it can only reach objects within an acceptable range. Servo's do not move in degrees, but a reasonably accurate conversion of roughly 9 servo points to represent 1 degree can be used for adjusting the yaw. However the weight, speed, and the slight variability of the servo sensors does cause some error, which is increasing noticeable the closer the arm gets to its maximum reach.

The arm is placed on a table, which in this report is called the operation table. The arm is connected to a host pc through a serial interface device. To control the arm, packets which contain information regarding the degree of movement for each servo motor and speed of each joint, are sent to the interface device. The SC-322 servo controller was the Interface device which contained an ATMEGA8 microcontroller, which generates pulse width modulation (PWM) square waves to control the servo motors.

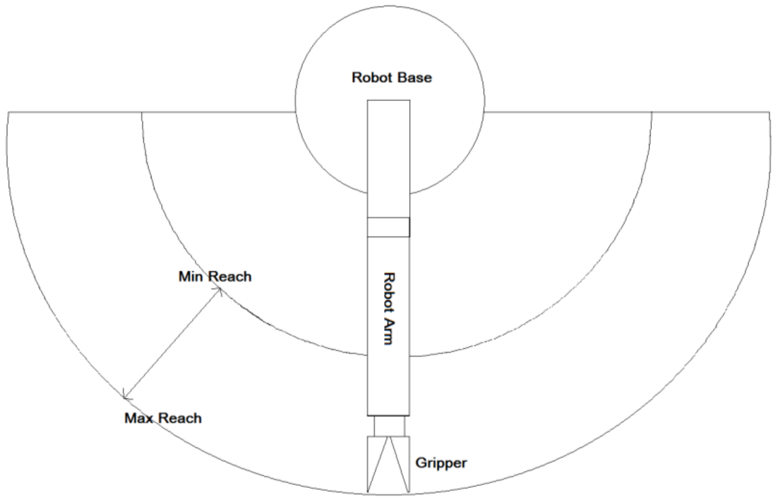
EMG data and vision data are fused with each other and are used to control the arm. Vision determines position of the object in the human working area, and sends command, to the robotic arm to move it to the desired position.

At the beginning of the program an initialization function is executed which sets the arm to starting position. The starting position is constant and sets the arm in same position every time the program is executed. To control the arm, based on position of the object in the human working area, row and column of the object in image should be converted to yaw and pitch of the robotic arm. The following is the equation to convert row and column in the image to yaw and pitch of the robotic arm:

$$yaw = \begin{cases} -\dfrac{column - \dfrac{imageWidth}{2}}{2} , & column > \dfrac{imageWidth}{2} \\ \left\| \dfrac{column}{2} - 80 \right\| , & column \leq \dfrac{imageWidth}{2} \end{cases}$$

$$pitch = minPitch + (row \times 4)$$

Where Column is the number of column of the object in image, and row is the number of the row of the object in image. By the above equations, position of the object is mapped to yaw and pitch for the robotic arm movements. Yaw and pitch are used to control the position of arm in 3D space. The gripper is controlled by the EMG directly, and based on the EMG data, it can be opened or closed.



Integration

This system is combined from three parts. Vision system, EMG and arm control, which all require huge resources of memory and processor time. At development stage of this project, we noticed that the EMG and the vision programs conflicted with each other, and because of the resources that they used simultaneously and the responsiveness they both required, they cannot be executed together. To solve this problem, after a few discussions, it was decided to use multithread programming techniques in the main program to manage all parts of the system. The EMG and vision programs have separate threads that run concurrently. Arm control is embedded inside the vision program. An investigation carried out to find bugs common in multithread programming (such as starving & etc) and was tested several times for assessing the performance.

Future Work

The work done so far proves that collaboration between EMG and Vision is beneficial. However there is still a lot of future work that can be done on the project. Firstly, if more complex objects manipulation is required then it is likely that more EMG sensors will be needed for complex operations i.e. for more classes of gestures. It is also necessary to determining the optimum EMG\Vision fusion for best performance, currently EMG is only partially used, and it is worth investigating replacing the gripper with a robotic hand and then using a greater portion of EMG for control, this would allow the robot to handle a variety of object shapes and sizes. The addition of pressure sensors would also benefit the system as it would be able to determine the level of force required to hold an object. There is the argument for better equipment and more sophisticated algorithms, as a bigger better robot with enhanced software will be able to handle larger object in an efficient manner, but is does depend on the final implementation as to what would be the best. The future direction of the project is down a ‘‘teaching by showing” route. Whereby a user will perform some action and the robot will learn that motion and be able to perform it. This has numerous applications, but would be especially useful for adaptive home robots. You could teach the robot how to cut a tomato, peel an onion or even to make your bed and fluff the pillows. We learn a majority of our skills from being taught; therefore using the same techniques you could also teach a robot.

Flowcharts