Recognition of Arm Gestures Using Multiple Orientation Sensors: Gesture Classification

Jean-Christophe Lementec and Peter Bajcsy

Abstract—**We present a gesture recognition algorithm from Euler angles acquired using multiple orientation sensors. This algorithm is a part of a system for controlling Unmanned Aerial Vehicles (UAVs) in the presence of manned aircrafts on an aircraft deck. After exploring multiple approaches to arm gesture recognition, we investigate a real-time arm gesture recognition system using the IS-300 Pro Precision Motion Tracker by InterSense. Our work consists of (1) analyzing several gesture recognition approaches leading to a selection of an active sensor, (2) gesture modeling using Euler angles, (3) low-level gesture characterization, and (4) model-based gesture classification algorithms. We have implemented and tested the proposed real-time arm gesture recognition system in a laboratory environment with a robot that represents an UAV surrogate**

I. Introduction

WE address the problem of gesture recognition for controlling unmanned and manned vehicles without interfering with the current control mechanisms of manned vehicles, such as, Navy [2] or NASA [15] aircrafts. Our objective is to explore multiple approaches to arm gesture recognition, and investigate a real-time gesture classification algorithm based on Euler angles as obtained from the IS-300Pro Precision Motion Tracker manufactured by InterSense [1]. A diagram in Fig.1 overviews our gesture recognition system, where a robot [3] serves as a UAV surrogate and four orientation sensors altogether are attached to upper arms and forearms of a flight director (2 sensors per arm).

In this work, we describe (1) our analysis of several gesture recognition approaches in Section II leading to a selection of an active sensor, (2) gesture modeling using Euler angles in Section III, (3) low-level gesture characterization in Section IV, and (4) model-based gesture classification algorithms in Section V. We conclude in Section VI with our observations about the robustness and deployment of the proposed gesture classification system.

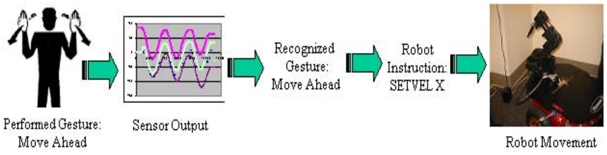


Fig. 1 Flow diagram of a developed system for robot control using

II. Arm Gesture Recognition Approaches

A. Overview of Gesture Recognition Approaches

Arm gesture recognition can be approached with using active or passive sensors, or a combination of both sensor types. An example solution using passive sensors would be a vision-based system [4],[13],[14]. Single or multiple cameras acquire video stream that is processed and gestures are mapped into temporal signatures of changes in video frames [4]. This solution faces several challenges in such a harsh environment as the aircraft carrier deck and has to overcome changes in a flight director orientation, outdoor illumination (day and night), and possible occlusions of flight directors or recognizing the right flight director among many directors on the deck. On the other side, this approach does not require any changes in the current control practices, or any changes in the flight director's equipment. One should be aware during a system design that any additional weight to the equipment worn by flight directors would increase fatigue of flight directors and hence additional weight is not desirable. This consideration imposes real-world constraints on systems with active sensors since they have to be worn.

Examples of solutions using active sensors would include gloves with bent sensors [7] or miniaturized accelerometers [8], [9]. For example, the cyberglove in [7] uses 18 distributed bent sensors embedded in a glove to capture finger articulation. Similarly, the advancement in Micro-Electro Mechanical Systems (MEMS) led to building a glove prototype at UC Berkeley [9]. Most of these solutions have been developed for indoor virtual reality (VR) applications and are not easily extensible to outdoor applications with highly uncontrolled environment. In addition, outdoor applications might require a feedback mechanism since a controlled vehicle can be out of sight [5], [6].

Manuscript received on March 31, 2004. This work was supported in part by the U.S. Navy under Grant N00014-03-M0321.

J. C. Lementec is with CHI Systems, Inc., Fort Washington, PA, USA

P. Bajcsy is with the National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign, IL 61820 USA. (Corresponding author phone: 217-265-5387; fax: 217-244-7396; e-mail: pbajcsy@ncsa.uiuc.edu).

0-7803-8500-4/04/$20.00 ©2004 IEEE

The use of passive and active sensors together was reported in the past [11] with the goal of combining advantages of both sensor types. For instance, placing fluorescent markers on tracked objects and illuminating them with known light sources is an example of a vision-based hybrid system that does not constraint moving subjects with heavy or bulky sensors and improves robustness of a standard vision based system in terms of motion detection and tracking.

We should also mention that the specific problem introduced in this section could have been also approached by broadcasting video of synthesized gestures to the cockpit of manned aircrafts. A computer program driven by a flight director would create video of synthesized gestures. Pilots of manned aircrafts would recognize synthesized gestures the same way as they did in the past, and all unmanned vehicles would receive directly the de-coded (interpreted) commands. We developed video examples of synthesized gestures for test purposes. However, this solution, although very robust from gesture recognition viewpoint, is not acceptable by the end application because the person giving commands has to be present on the aircraft deck during the entire time of any vehicle navigation.

B. *Proposed Approach and Sensing*

While there are many approaches to gesture recognition, we chose to research and develop a solution with active sensors because of the end application requirements on performance robustness and reliability. By considering the importance of (a) system reliability in a highly varying environment (e.g., geometry, illumination, line of sight, temperature, and operator's fatigue) and (b) safety of navigation operations, the active sensing approach outperforms solutions based on passive sensing approach. As one part of our research, we surveyed and evaluated active sensors based on their (a) size, (b) weight, (c) cost, and (d) commercial availability. We considered three different solutions, such as, (1) virtual reality (VR) motion trackers [1], (2) global positioning systems (GPS) [10]] and Micro-Electro-Mechanical Systems (MEMS) with tiny operating system (tinyOS) [8], [9]]. The choice of the IS-300 Pro Precision Motion Tracker by InterSense, MA, for this work was primarily driven by its best sensing performance specifications and its commercial availability. For example, a spatial accuracy of GPS (around 3 m for the GPS with the Wide Area Augmentation System) and an extra development effort (building a glove with MEMS sensors) were considered as major drawbacks of the other two solutions. The cost of IS-300 Pro Precision Motion Tracker ($4,375 for the base unit plus $1,437 for each additional sensor), and the size and weight parameters (each sensor cube weighs 2.1 oz and measures 1.06”x1.34”x1.2”) were at the borderline of being acceptable at the time of purchase. Nevertheless, the vendor has miniaturized the sensors and decreased their weight significantly since the time of purchase.

Given the choice of an active sensor, our approach to the problem of gesture recognition is based on (1) translating arm motion into a temporal sequence of orientation of angles, (2) describing a sequence of orientation angles with its characteristics, (3) building models of gestures in a lexicon using sequence characteristics of orientation angles, and (4) classifying sequences of orientation angles into gesture classes according to the developed gesture models in real time. The basic premise of our approach is an existence of a unique mapping between human gesture represented by arm movements and a temporal sequence of upper arm and forearm orientation angles. This type of mapping is frequently used in the computer graphics community, where arms are modeled as connected cylinders or ellipsoids, changing their orientation in a world coordinate system. Our overall approach is fundamentally robust to most environmental conditions on an aircraft carrier that makes the vision-based solution difficult. These conditions include variable lighting, occlusion in the line of sight, background clutter, fog, and hot engine exhaust. Distance from the director to the aircraft is not a factor either as long as the communication between the yellow shirt and a specific aircraft (manned or unmanned) can be established. Communication is clearly a problem, but our system requires very low bandwidth (only communicating high level commands at a frequency less than a few hertz). We assume that the generality of the proposed solution would be validated in multiple application environments by a study (similar to [12]) and specific modifications would be performed accordingly.

III. GESTURE MODELING USING EULER ANGLES

There are many different ways to measures orientation in a three dimensional space. It all depends of the referential system being used. For our application, we found that the Euler coordinate system was the most suitable. This system represents an orientation with three different angular values: yaw, pitch and roll. These values are commonly used to describe the orientation, or attitude, of an aircraft as shown in Fig. 2. Roll indicates rotation along the front-to-back axis of the plane; Pitch indicates rotation along the side-to-side axis of the plane; Yaw indicates rotation along the vertical axis (or the axis perpendicular to the other two, if the plane is not level).

When a sensor is attached along the side of one arm (or forearm), the roll axis is along the length of the arm, the pitch axis is horizontal and perpendicular to the roll axis, the yaw axis is vertical. As the roll axis always “follows” the orientation of the arm, it provides a relative angle (to the arm) for both the pitch and roll. For the sensors on the forearms near the hands, it is indicative of the orientation of the hand. For example, palm facing back or front, or facing up and down. The pitch angle indicates the angle relative to a horizontal plane. Having one's arm horizontal will provide an angular value of 0; and vertical will lead to a value either +90 or –90 depending on whether the arm is oriented toward the ground or the sky. Yaw values are indicatives of the compass orientation of the arm (North, West, South or East). Unlike the other two angles, yaw angle values depend on the absolute orientation (relative to the earth) of the sensor wearer and hence cannot be directly used for robust classification. Yaw can only be used for low-level pattern characterization (oscillation or steady) or as a relative value with respect to a yaw value from another sensor.

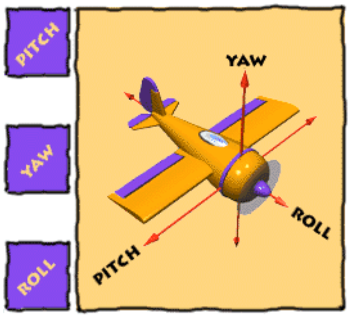


Fig. 2: Roll, Pitch and Yaw axes.

A. *Bouncing Effects and Discontinuity*

While Euler angles are easy to understand and model, they introduce temporal discontinuity of values. Next, we describe gesture modeling related characteristics of each angle separately. First, pitch angles always range between -90 and plus 90 degrees. However, when passing the 90-degree vertical point during an arm rotation, the pitch values will be decreasing again. Yaw values, however, have a problematic discontinuity when passing through the 90-degree position. At that point, the yaw value will suddenly shift by 180 degrees. For example, if the movement was started with a North heading, it will shift to South heading when passing the vertical point. Fig. 3 illustrates this transition and the curves S1.1 and S1.2 correspond to the yaw and pitch values respectively. The movement was an arm oscillation from about 0 degree pitch (horizontal) to 30 degree beyond vertical. The yaw angles represented by curve S1.1 show the 180 degree jump at the time of the transition past vertical. When reaching -90 degrees, the pitch curve S1.2 reverses its course for the remaining 30 degrees. The "bouncing effect" (or reversed course) of pitch values observed in Fig. 3 may yield to a misinterpretation of the actual movement frequency due to oscillations centered on the vertical position. Fortunately, it is possible to use the yaw 180-degree shift information to determine when such effect is occurring and adjust the pitch value by "Corrected pitch"="raw pitch" + 2*(90-"raw pitch").

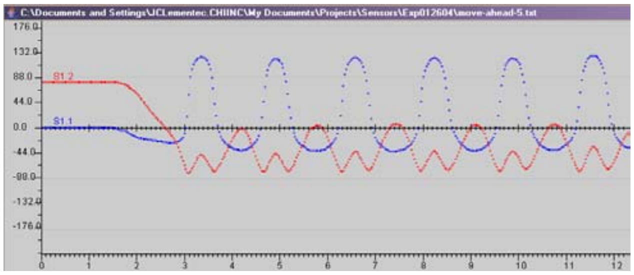


Fig. 3: An example of a bouncing effect due to oscillations past the vertical orientation.

B. *Rollover Effect*

Another problem occurring while using Euler angles is the rollover effect. The angular values provided by orientation sensors are bounded by -180 and +180 (except for pitch values which are between -90 and +90). When an angle value goes past 180 degrees, then it is kept within a range of [-179,181] degrees by subtracting 360. Fig. 4 and Fig. 5 show observed oscillations around the rollover point with no corrections and with the corrections by subtraction. The rollover correction is practical since we do not expect a complete rotation of a human arm.

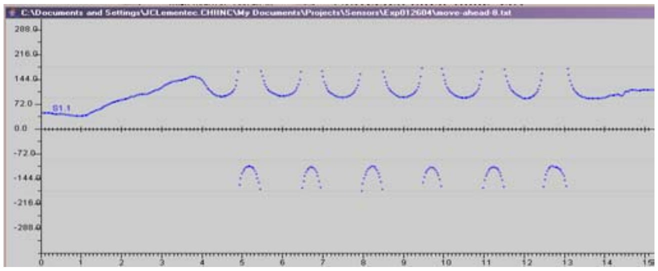


Fig. 4: Oscillations around the rollover point with no corrections.

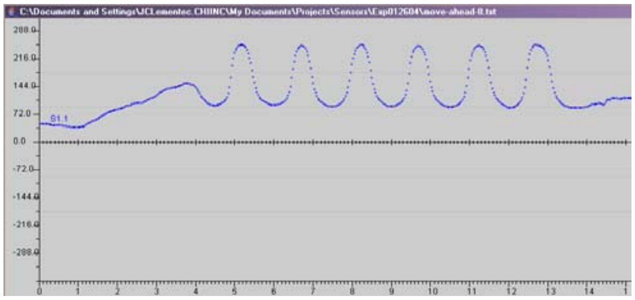


Fig. 5: Oscillations around the rollover point with corrections.

Our correction is also based on the assumption that there is no rate of changes greater than 180 degrees per 10 ms, and both raw and corrected values are available for classification purposes. The correction routine is presented below.

```
get_corrected_angle(previous_raw, previous_corrected, current_raw )
    {
    corrected_raw = current_raw;
    if (abs(current_raw-previous_raw) > 180 )
        if (current_raw < 0)
            corrected_raw = current_raw + 360;
        else
            corrected_raw = current_raw - 360;
    return previous_corrected + (corrected_raw - previous_raw);
    }
```

IV. Low Level Gesture Characterization

Our approach to robust gesture recognition relies on a two-stage classification technique. The first stage characterizes temporal streams of each Euler angle separately. The second stage uses the combination of Euler angle stream characteristics from the first stage to assign gesture labels according to a set of gesture classification models.

As each of the 4 sensors reports 3 Euler angles (yaw, pitch and roll), we receive 12 temporal streams of angular values. The stream information content is contained more in the evolution of values over time than in the instantaneous value of an Euler angle. We found two basic patterns to characterize an evolution of angular values, denoted as steady and oscillating patterns. Patterns that could not be identified as either steady or oscillating are tagged as unclassified. Many gestures involve some back and forth movements of one or both arms. The orientation sensors report these movements as a sinusoidal modulation of one or more angular values, and we labeled those patterns as oscillation. Most gestures also involve some holding position of one or both arm. These measurements are represented as a flat curve for the related angular values, and we labeled those patterns as steady.

At any moment each of the 12 streams of angular values are labeled as, steady, oscillating or unclassified. In addition of this categorization, a position value was also determined for the steady and oscillating state, with five different possible positions:

HIGH	: median value < -67.5
MEDIUM-HIGH	: -22.5 < median value < -67.5
MEDIUM	: 22.5 < median value < -22.5
MEDIUM-LOW	: 67.5 < median value < 22.5
MEDIUM	: median value > 67.5

A. *Detecting steady state*

The purpose of a steady state is to characterize a holding position. As the yellow shirt may not be able to hold a perfectly still position after a long working shift, the algorithm must be sufficiently tolerant. Small shaking of the arms should not be labeled as oscillations, and slow drift of the position should be tolerated.

We experimented with different estimations of gesture speed and acceleration by using low pass filtering but eventually settled in for the following simple solution. A data stream of angle values was considered steady if there was a variation of no more than 18 degrees for at least 0.4 seconds. These values were determined empirically from experiments. The delay of 0.4 seconds seems to be the minimum amount of time that a position would be held.

B. *Detecting oscillation state*

For the detection of oscillations, we used a technique of minima and maxima detection. Any data stream can be characterized as a sequence of alternating minima and maxima. If the stream is steady, then the difference between two consecutive minima and maxima is small which is opposite to oscillations. To accommodate for noise perturbation, any min/max or max/min pair were rejected if (max-min) was less than 10 degrees. Thus, an oscillation is detected (1) if there is a sequence of min/max/min or max/min/max where the time difference between the first min (or max) and last min (or max) was at least 0.3 seconds but no more than 2 seconds; (2) if (max – min) was at least 35 degrees; and (3) if the difference between the two minima (or maxima) was no more than 40 degrees. Anytime such a sequence is detected, the pattern is labeled as oscillating. Note, that there is a detection delay of at least one period after the beginning of the oscillation. Another limitation to our current algorithm is that there is a trailing effect. Oscillation detection takes precedence over steady state detection and therefore we will wait for the maximum allowed duration of one period (2 seconds in our current setting) before we acknowledge the end of the oscillation.

This algorithm was adequate for our experiments but could be improved to reduce the latency and trailing effect. The maximum allowed duration of a period (2 seconds) is also probably too long and could be readjusted.

V. Gesture Classification and experiments

The actual gesture classification is based on the low level characteristics of individual angular data stream. Whenever new low-level characteristics are detected, the gesture classifier is activated. By using the set of current angular characteristics, a gesture label is determined based on built-in gesture model. This approach does not need to detect the transition from one movement to another. It simply compares the current combination of low-level characteristics with a pattern (expressed as a logical formula) for each of the possible 20 gestures to be recognized. The classification algorithm does not make any assumption about the possible order in which the gestures could occur. If such preferential gesture order exists in practice, then this information may be used to limit locally the number of patterns that must be tested at a particular time. This a priori information could potentially help in disambiguating some gestures, but it did not appear to be necessary based on the data we have collected.

In our current configuration, we are able to recognize very robustly those 11 out of 20 gestures that do not need the yaw angle information. Yaw values are dependent on the orientation of a flight director and therefore, cannot be used individually but only in comparison with another yaw value. Robust recognition of the remaining 9 gestures requires incorporating modeling enhancements and inclusion of the yaw angle.

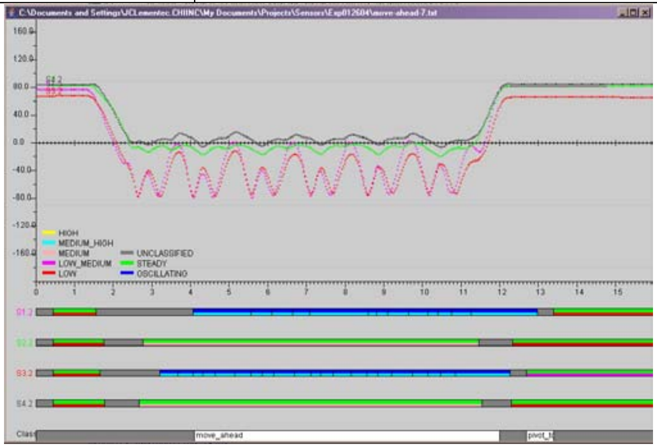
We present next the results obtained and the logical formula used for pattern modeling of each of the NAVY gestures [2]. We represented the low level classification of each individual data stream as color-coded bars. Unclassified states are represented as a grey bar. Steady and oscillating state bars are divided in two parts where the upper part indicates steady or oscillating and the lower part indicates the position

Abbreviations for the formulas are as follow:

S1.1	lafy : left arm front yaw (fore arm)
S1.2	lafp : left arm front pitch
S1.3	lafr : left arm front roll
S2.1	laby : left arm back yaw (upper arm)
S2.2	labp : left arm back pitch
S2.3	labr : left arm back roll
S3.1	rafy : right arm front yaw
S3.2	rafp : right arm front pitch
S3.3	rafr : right arm front roll
S4.1	raby : right arm back yaw
S4.2	rabp : right arm back pitch
S4.3	rabr : right arm back roll

A. Move Ahead

8 MOVE AHEAD	Arms extended from body and held horizontal to shoulders with hands upraised and above eye level, palms facing backwards. Execute beckoning arm motion angled backward. Rapidity indicated speed desired of aircraft.



lafp.oscillating(MEDHIGH) AND
(labp.steady(MED) OR rabp.steady(LOWMED)) AND
rafp.oscillating(MEDHIGH) AND
(rabp.steady(MED) OR rabp.steady(LOWMED)))

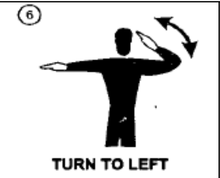
Fig. 6. Description of “Move Ahead” gesture.

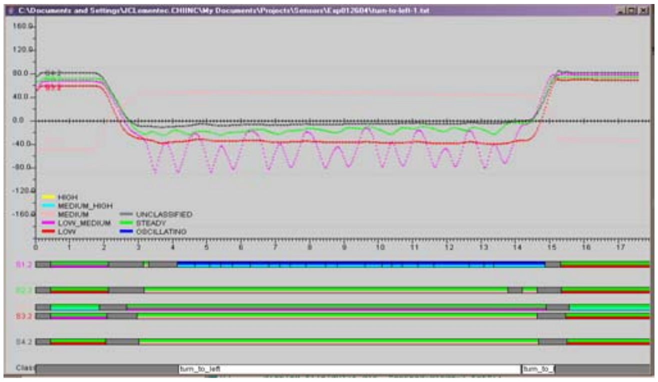
For the “Move Ahead” gesture, the pitch measurement shows both forearms oscillating around the medium-high position. Upper arms only have parasitic motions and are characterized as steady since the values stay within 18 degrees of motion margin. The pattern specifies the median position of the upper arms as either medium or medium-high.

The trailing effect of oscillation is causing in this case a false classification of pivot to left. This problem will disappear once we have a better algorithm for detecting oscillation. This problem has occurred regularly in our experiments and it affected all movements involving an oscillation.

The upper arm characterization reported some times medium position and other times medium high position. Our pattern includes both positions as valid. The parasitic oscillation of the upper arms was also more or less pronounced from one user to another, though always within the tolerance for a steady state.

B. Turn to Left

	Extend right arm horizontally, left arm repeatedly moved upward. Speed of movement Indicates the rate of turn. A clenched fist (day) or down-turned wand (night) directs the pilot to lock the indicated brake.

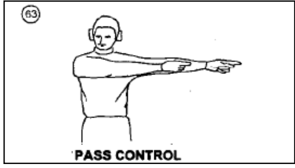


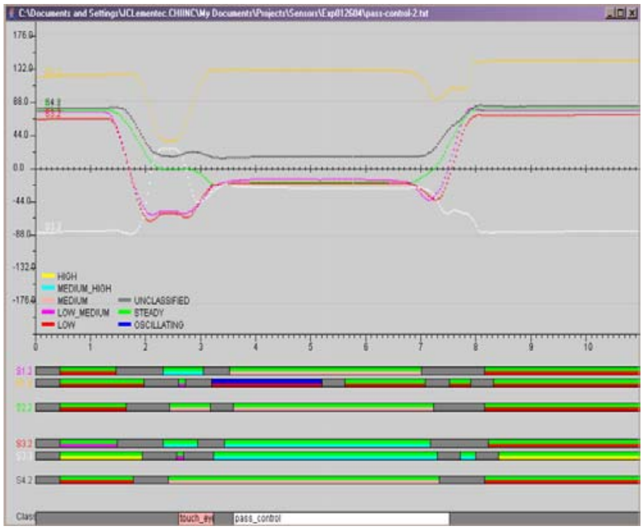
lafp.oscillating(MEDHIGH) AND
labp.steady(MED) AND
(rafp.steady(MED) OR rafp.steady(MEDHIGH)) AND
rabp.steady(MED))

Fig. 7. Description of “Turn to Left” gesture.

In this movement we observe the same pitch oscillation of the left forearm as for the “Move Ahead” gesture. In this case, the right arm remains steady and horizontal. Note, that the pattern also accepts a medium high position for the right arm. The yaw value of the right arm (curve S3.1 in Fig. 7) indicates the direction in which the arm is pointing. In this case again, both upper arm characteristics appear unnecessary. They could be replaced by the difference in a pair of yaw measurements.

C. Pass Control

	With both arms shoulder height in direction of person receiving control. Will touch eyes or hand with both hands first before pointing.



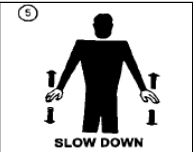
Touch Eyes
(lafp.steady(MEDHIGH) || lafp.steady(HIGH)) &&
(rafp.steady(MEDHIGH) || rafp.steady(HIGH)) &&
(lafr.steady(LOWMED) || lafr.steady(MED)) &&
(rafr.steady(LOWMED) || rafr.steady(MED))

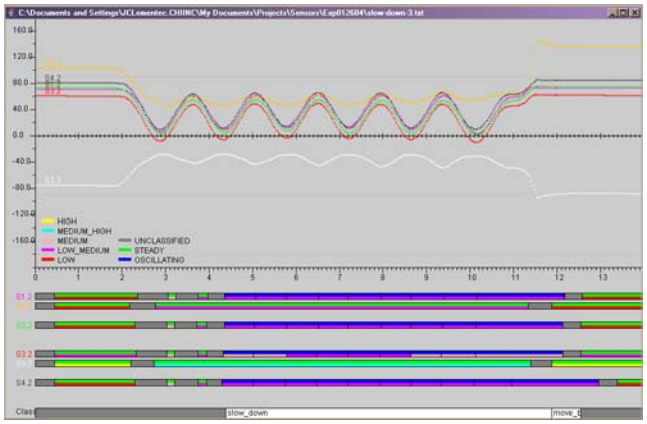
Pass Control
(lafp.steady(MED) || lafp.steady(MEDHIGH)) &&
(labp.steady(MED) || labp.steady(MEDHIGH)) &&
(rafp.steady(MED) || rafp.steady(MEDHIGH)) &&
(rabp.steady(MED) || rabp.steady(MEDHIGH))

Fig. 8. Description of "Pass Control" gesture.

This gesture was interesting as it involved two sub-gestures: touching eyes and then pointing toward another flight director. In this case, the pattern requires the specific sequence of one sub-gesture followed by the other one (not expressed in this formula). Note, how the roll value is being used to detect the tough eyes pattern

D. *Slow Down*

	Arms down with palms towards ground. Then moved up and down several times.



(lafp.oscillating(LOWMED) || lafp.oscillating(MED)) &&
(rafp.oscillating(LOWMED) || rafp.oscillating(MED)) &&
lafr.steady(LOW)!= true && rafr.steady(HIGH)!= true)

Fig. 9. Description of "Slow Down" gesture.

Both upper arms and lower arms are oscillating with same amplitude. We actually found some differences in measurements where in some cases the upper arm was not oscillating very much but we did not include them. The most important factor is the low-medium to medium position of the oscillations (by contrast to high-medium for "Move Ahead" gesture).

VI. Conclusion

We presented a real-time gesture classification system using multiple orientation sensors that has been tested for robustness and speed. Based on our gesture recognition analysis, we concluded that by incorporating the yaw angle and enhancing the current model, we could eliminate the upper arm sensors, which would lead to weight and cost reduction of the whole system. Robustness of some specific gestures that include opening and closing fists (brakes gesture) might be improved by additional information.

References

[1] Intersense web site: http://www.intersense.com/

[2] US Navy, "Field Manual FM1-564 Appendix A", web site: http://www.adtdl.army.mil/cgi-bin/atdl.dll/fm/1-564/AA.HTM

[3] ActivMedia documentation for ARIA, ActivMedia support web site: http://robots.activmedia.com

[4] Bobick A.F. and J. W. Davis, " The Recognition of Human Movement Using Temporal Templates," IEEE Trans. On PAMI, VOL. 23. NO. 3, March 2001, pp. 257-267

[5] Fong T., "Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation," Ph.D. Dissertation, CMU-RI-TR-01-34, November 2001 (156 p).

[6] Ferworn A. and W. Lu, "Optimization for Video and Telerobotic Control on Palm OS PDAs,"

[7] Kessler G. D., L.F. Hodges, N. Walker, "Evaluation of the CyberGlove as a Whole Input Device,"

[8] Smart Dust Project at http://www.cs.berkeley.edu/~awoo/smartdust/

[9] Hollar, S. E., "COTS Dust," Master Thesis at University of California, Berkeley, Fall 2000.

[10] Global Positioning Systems, The GPS Store, http://www.thegpsstore.com/site/

[11] Moeslund Thomas B., Granum Erik, "A Survey of Computer-Vision Based Human Motion Capture," Computer Vision and Image Understanding 81**,** 231–268 (2001).

[12] Cheng, V. H. L., V. Sharma, and D. C. Foyle, "Study of aircraft taxi performance for enhancing airport surface traffic control," *IEEE Trans. Intelligent Transportation Systems*, Vol. 2, No. 2, June 2001.

[13] Krahnstoever, N., M. Yeasin, R. Sharma, "Automatic Acquisition and Initialization of Kinematic Models," IEEE Conference on Computer Vision and Pattern Recognition, Technical Sketches, Kauai Marriott, Hawaii, USA, Dec, 2001.

[14] Gavrila D. M., " The Visual Analysis of Human Movement: Survey," Computer Vision and Image Understanding: CVIU, Vol. 73, No. 1, 82-98, 1999.

[15] Taxiway Navigation And Situation Awareness System (T-NASA) web page: http://human-factors.arc.nasa.gov/ihi/hcsl/T-NASA.html.