Angular Estimation of Human Motion Using Combined Gyroscope Sensor and Image Based Observations with Kalman Filter Approach

***Abstract* - Common user control interfaces are exemplified by translations of user interaction into an unnatural medium, such as joystick controllers or vehicle controls, requiring experience to use properly. In this work we propose a control device based on intuitive user motion interaction, where the device control consists of a common camera phone, and the user motion is replicated on a two degree-of-freedom camera mount system. We begin by building and testing the accuracy of a combined type of prediction algorithm, one based on images taken by the camera phone, and one using the sensors contained in the phone itself. The accuracy of each method as a function of angular velocity is determined and from these functions the prediction basis is divided into slow for the image based approach, fast where the sensor based approach is superior, and intermediate, where a Kalman filter approach combines the predictions of both.**

**Keywords: Robot Motion, Intuitive Control, Sensor Fusion, Kalman Filter, Image Based Motion Estimation.**

I. Introduction

Intuitive human robot interaction, where human subjects use natural modes of speech or gesture, remains a relatively unexplored area of robotic control. Traditional industrial robotic systems rely on constrained settings and precise closed loop programming to accomplish work tasks, even with recent advancements [1, 2, 3]. Another familiar form of robot are those that operate in the domestic domain [4, 5, 6], such as robotic cleaners. Typically, these robots can only be passively programmed to avoid stationary obstacles or stop working whenever they encounter uncertain environments, and thus cannot actively interact with or be controlled by humans in any meaningful way. In a broader sense both of these robots belong to the closed loop autonomous category, where they are assigned some task which cannot be dynamically redefined or solve new issues in their task space.

More recent advancements in robotic control and interaction rely on input from human users to determine the system heuristics. In contrast, traditional user system controls are simplified versions of human input, such as buttons or joysticks, which are limited to the non-intuitive actions that correspond to these inputs. The main issue with such systems is the required training and adjustment to control designs, as well as the lack of carryover from system to system, or game to game [7]. Intuitive control instead depends on natural motion that allows the human to operate in ways that they would understand relatively quickly with a little training or testing with the system in order to be able to utilize properly.

Treating the human and device as a combined dynamic system leverages natural proprioceptivity. Thus, estimation methods suitable to dynamic systems are warranted. This is of great interest in fields such as virtual and augmented reality, where intuitive control can correlate directly to realistic action. Such systems utilize natural gestures as opposed to buttons or joysticks or other forms of simplified control to make the experience of the user feel natural. Gesture-based human robot interaction is a developing field [8] of great interest to roboticists and human robot teams. In the future, robotic systems and users must be able to collaborate intuitively and simply in the evolving workplace [9]. Recently many different types of available new gesture-based technologies, inspired by the augmented or virtual reality systems [10, 11], where capturing natural user movement is fundamental to the system. These types of user control systems often appear as sensor harnesses where different sensors are attached to the user and the different readings are recorded and calculated to determine user intent.

Accelerometer gyroscope combinations such as those utilized in [12, 13] are the most common type of sensory systems that utilize human motion to determine user intent. By sensing the acceleration and rotational velocity experienced by each sensor, the underlying system mechanics and motion can be determined. Very precise accelerometers and gyroscopes can determine these readings with a very high degree of accuracy. Because such sensor outputs are based on actual mechanics, the motion can often be more explicitly modelled when compared to other methods. One drawback to this kind of system is it has no reference to the environment, only relative motion, and drift can play a large part in the error of the motion estimation.

When used with other inputs, accelerometer gyroscope combinations can render accurate renderings of motion and intent, especially when measuring muscular signals either mechanically or electrically. Electromyographic (EMG) sensors are another type of common application in human robotic interaction [14, 15]. EMG sensors measure the actual voltage output for muscle groups where they are attached and these measurements can be translated into input signals for user control, often using machine learning methods to estimate the system mechanics. Due to differences in user body composition, motion tendencies, and muscular utilization, it can be difficult to get precise or consistent outputs from these types of sensors, and often they are utilized in tandem with other types, after the effort of calibrating them to a particular user [16, 17].

In robotic control, image-based techniques allow systems to make use of the environment in the surrounding feature information, similar to how human vision operates, identifying objects or specific points in those scenes and using those to track objects or surroundings. Many computational image processing techniques rely on simpler versions of these, such as reducing tracking to specific keypoints, and sometimes processing these keypoints in applications such as facial recognition [18]. In more recent modern systems deep learning approaches are utilized instead [19]. The disadvantage to this type of deep learning approach is that excessive processing can severely hinder real time application. Despite this flaw, these programs have the capability of tracking entire objects and separating them from the surrounding environment. Simple keypoint detection, by contrast, allows fast computation and nearly real time application, and these keypoints can be tracked from frame to frame.

The system introduced in this work is a step in realizing intuitive, gesture based control on a common smartphone. The system is thus low cost, as the estimation is realized on an android app. Further, the system integrates the relative mechanical sensing capabilities of an inertial measurement unit (IMU) with the absolute position sensing by way of keypoint matching on a monocular camera. The keypoints are tracked frame to frame to determine the angular rotation. Sensor fusion by way of Kalman filtering is utilized on-line by estimation of a Gaussian distribution fit using obtained data which is then processed off-line. The system is agnostic to environment, as a set of reliable keypoints can be found in any such environment with sufficient lighting.

The system introduced here consists of three main components: a camera phone which is used to capture the motion of the user, a computer which receives the signals from the phone and records estimates the motion, and a 2 degree-of-freedom (DoF) mount operated by servo motors which replicates the angular rotation of the user with the camera phone. In summary, the remotely mounted camera should move as the user with the camera phone does, creating an intuitive, remote video stream system.

.

The robotic system makes use of multiple different sensing types and once combined they operate in a manner that reduces overall error in the motion estimation and robust estimation in multiple different movement and environment types, such as rotation speed or poor lighting or keypoint existence. For example, image processing tracking works well with slower rotational movement, where the image resolution can render a high degree of accuracy with keypoint tracking. In other situations, such as faster rotation, sensors such as accelerometers and gyroscopes can render a better estimation of movement. By considering the advantages and disadvantages of both systems, the combined system can better operate in a range of different conditions that it would otherwise be unable to by relying only on one source of estimation or the other, and be far more prone to failure as a whole where accuracy is an important system aspect.

Error in image estimation is a function of two features, processing rate and the approximation of the curve on a spherical surface. As the angular change between frames increases, the distance approximation along this curve diverges from the length of the curve itself, and the angle is then underestimated. Additionally, error in keypoint matching increases as the angular view changes, and it becomes more difficult to find reliable matches. Overall, the error in image estimation can be expected grow exponentially as the angle between frames increases, eventually being unable to find matches and failing completely. Sensor based angular change estimation, by contrast, can be expected to decrease in a negative exponential curve, as the signal to noise ratio increases well above the noise of a slow or still system. Since both systems perform well under opposite rotational velocities, a combined approach utilizing both methods can compensate for the weaknesses of both.

Kalman filters have been utilized for several different types of estimation where relying only on measurements can create significant uncertainty in finding the actual system state. For example in [23], Kalman filters represent a application of a Markov chain, where the approximation of the previous state and the expected measurement error combined with the current expected measurement error can render a more accurate estimation of the current state. Another application of the Kalman filter such as [24] takes the form of sensor fusion, where multiple different measurements are combined by weighting their expected error and forming a new average estimate of the state. This application of the Kalman filter is utilized in this work to combine sensor and image based estimation of rotation.

This work is divided into the following sections: I. Introduction to the system goals and underlying concepts, II. Design of the system and Methods utilized, III. The experimental setup, procedure, and results of the experiment, and finally IV. Concluding remarks.

II. SYSTEM DESIGN AND METHODS UTILIZED

*2.1 Overview of the System*

The main function of the design is to serve as a type of remote point of view system, what this means physically is that the motion of the user with the camera phone is re-created on the camera mount system which is controlled by two servo motors along two different axes, a vertical axis and the lateral yaw axis. The entire system consists of three different components, the first of which, the camera phone, serves as the interaction point between the user and the system. The computer potion of the system serves to receive the incoming prediction and to send the commands to the camera mount portion. The camera mount component, which is attached to a control board, serves as the actualization element, which reproduces the user motion using the two servo motors. A simple visualization diagram for the system can be viewed in Fig. 1. The phone utilized in this system is the LGG 810 Q device which operates on an android platform. The computer is an ASUS GU502G Windows PC utilizing python scripts to process the incoming image and sensor data and sends the calculated angle to the Botboarduino controller, which then moves the two Hitec HS-85BB servo motors to replicate the motion of the user.

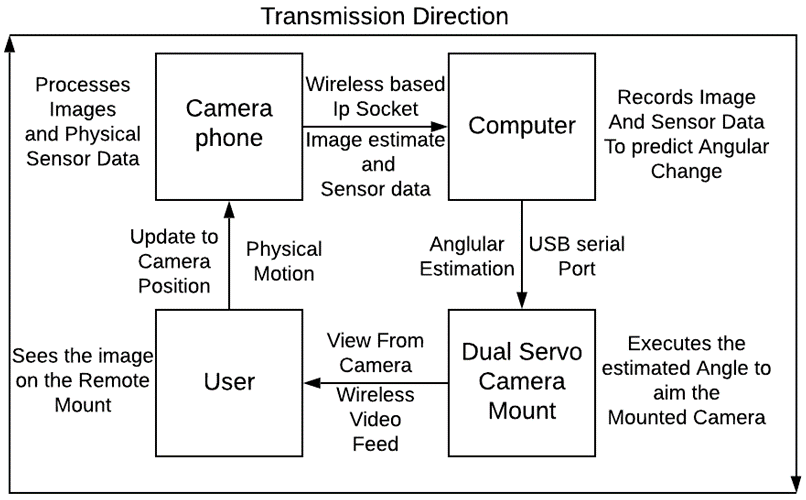


Fig. 1: System visualization

For communication between the various components, the Android phone and laptop communicate using TCP wireless protocol while both are connected to the same network using server-client sockets. The laptop and Botboarduino communicate using a serial port USB connection. The incoming sensor data consists of the linear accelerometer, gravitational vector, and gyroscope data, passed as a string which is decoded by the laptop. The laptop sends a request signal and a received signal that to keep the data from being interpreted only in single sets. Similarly, the image data is processed using the opencv java API, and the calculated angular changes are sent to the computer. The phone begins the process of taking the next image and calculating the angular change, until the signal from the laptop for the next image is received, then the process starts again.

*2.2 Image-Based Prediction*

For the image-based approach to angular prediction, the keypoint detection is utilized. Traditional forms of key points include edge detection and corner detection, but these key points do not work well when changes in lighting or viewing angle occur. Therefore, the scale invariant form of keypoints such as what was originally developed in the SIFT[20] and SURF[21] methods is performed on the image using different image resolutions to define key points in multiple scales. This method generates reliable matches as any points that appear at one scale but not at others are discarded int eh detection scheme. Images are converted to grayscale before processing. These scale invariant keypoints are more easily detectable and robust without regards to lighting or viewing angle, and in this application more easily trackable. Here we make use of the ORB algorithm [22], due to the reduced requirement of system resources as compared to other methods, which is important for faster tracking of motion. Higher sampling is desirable, as the change in angle is smaller and more locally Euclidean. The frequency of the processing is around 2.5hz, but keypoint processing and matching is dependent on the input images and so varies somewhat. The results of this type of detection can be seen in Fig. 2, where the keypoints and magnitude and orientation descriptors are drawn on a red panda image.

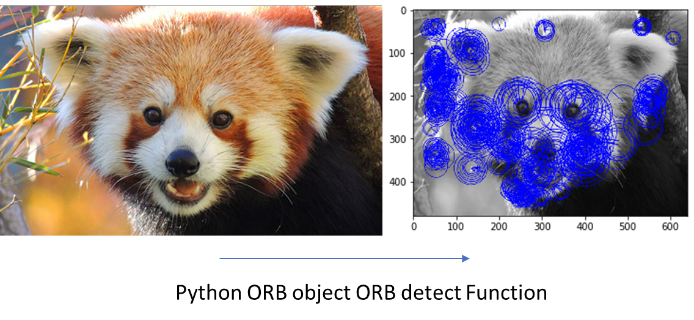


Fig. 2 Keypoint detection and description.

For sequential images, the keypoints are matched using the BFMatcher (brute force matcher) object and the relative motion in frame of the best 10 keypoints is calculated. The average motion of the keypoints is computed as the angular rotation when measured within the field of view of the camera, using Eq. (1):

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$\Delta\theta=\frac{Fov}{res}\frac{1}{N}\Sigma\Delta x\;\;\;\text{(1)}$

\end{document}

where field of view (*Fov*) is divided by the image resolution (*res*) and multiplied by the sum of pixel movement () and divided by the number of movement samples (*N*). This can be seen as the average pixel movement of each of the keypoints, which is then normalized as the angular change in the *Fov*. An example of this process can be seen in Fig. 3. The opencv matcher algorithm, however, is occasionally prone to bad matches, so the average and standard deviation are computed for the set, and outliers are removed, finally the average is recalculated. This method works well even when the image resolution is somewhat limited.

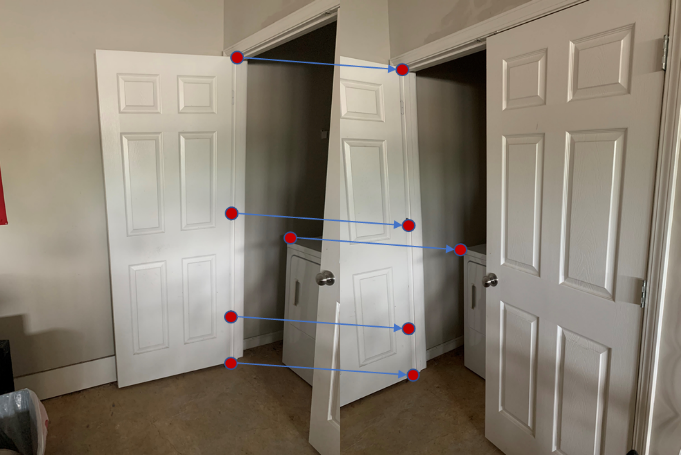


Fig. 3 An Example of keypoint matching.

*2.2 Sensor-Based Prediction*

Gyroscope sensors are widely used in modern smartphones, which can track the motion of the phone itself. In this paper, the gyroscope sensor is set in the GAME delay mode, i.e., output at 50Hz. All gyroscope data are relative to the phone frame shown in Fig. 4, which changes when the phone moves. To reduce the influence of noise, a classic Butterworth lowpass filter is applied on the sensor data before they are used for pose calculation. During each sampling interval, rotation velocity of a phone is considered constant. Given initial direction of the phone, the relative final direction can be calculated in the following way:

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$\theta_{1}=\dot{\theta}_{0}\Delta t\;\;\;\;\text{(2)}$

\end{document}

where , a three-dimensional (3D) vector, are initial rotation velocities, acquired from the gyroscope sensor, and ∆t is the corresponding sampling interval. For consecutively sampling process, initial position of the process is defined as the absolute reference frame, i.e., . When the first data comes in, Eq. (2) can be used to calculate the initial pose of the next sampling interval. For the second sampling interval, pose of the phone relative to the first sampling interval can be calculated using cardan angles

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$R_{1}^{\prime}=\begin{bmatrix}c_{z}c_{y}-s_{z}c_{x}s_{z} & -c_{z}s_{y}-s_{z}c_{x}c_{y} & s_{z}s_{x}\\
s_{z}c_{y}-c_{z}c_{x}s_{y} & -s_{z}s_{y}+c_{z}c_{x}c_{y} & -c_{z}s_{x}\\
s_{x}s_{y} & s_{x}c_{y} & c_{x}
\end{bmatrix}\;\;\text{(3)}$

\end{document}

where , , , ,, and .Thus, the ending pose of the second sampling interval can be defined in the absolute reference frame as . Due to the fact that , ***R***0 is an identity matrix. For the third sampling interval, the pose of the phone can be defined as . Where is the pose of the phone relative to the previous sampling interval. Iteratively, pose of the n-th sampling interval can be obtained by:

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$\begin{array}{c}
R_{n}=\begin{bmatrix}c_{z}c_{y}-s_{z}c_{x}s_{z} & -c_{z}s_{y}-s_{z}c_{x}c_{y} & s_{z}s_{x}\\
s_{z}c_{y}-c_{z}c_{x}s_{y} & -s_{z}s_{y}+c_{z}c_{x}c_{y} & -c_{z}s_{x}\\
s_{x}s_{y} & s_{x}c_{y} & c_{x}
\end{bmatrix}\\
R_{n}=R_{n}^{\prime}R_{n-1}=R_{n}^{\prime}R_{n-1}^{\prime}\cdots R_{1}^{\prime}R_{0}
\end{array}\;\;\text{(4)}$

\end{document}

where , , , ,, and . The prime indicates incremental of the angles between two samples. Based on Eq. (3), cardan angles of each sampling interval relative to the absolute reference frame can be calculated, which represent the current pose of the phone relative to the absolute initial pose. In addition, these angles are also exactly the absolute rotation angles of our actuator.



Fig. 4 Definition of phone frame.

*2.3 Combined approach*

For certain angular velocities where the error of both approaches is high, a Kalman filter based approach to both estimations is employed to rectify the uncertainty in this range. For this system the range was designated as 10-20 deg/s and the Kalman filter was developed to combine the two angular estimations from each method when the angular velocity fell into this range. The determination of the range will be demonstrated in the result discussion. The equation developed to handle the combining of the two estimation techniques can be seen as

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$\begin{array}{c}
x_{k}=x_{k-1}+A\Delta x\\
\Delta x=Cz
\end{array}\;\;\text{(5)}$

\end{document}

where the matrix *C* represents the weight given to each observation, *∆x* the combined observation, *x* the previous and current states, and *A* the matrix that transforms the measurement to the state estimation. To adjust for the difference in the frequency of the estimation data (the frequency of sensor’s data is higher than that of image data), the state change from the sensor-based method accumulates in time until an additional measurement of image data is available. At this moment the two are weighted and summed to create the new state change, which is added to the previous state estimation. To account for the effective ranges of the system, the matrix *C* varies dependent on the velocity estimation. Below the Kalman range it weights only the image measurement, *C* = [1 0], above only the sensor measurement *C* =[0 1]. Within the Kalman range the weights are calculated by the Kalman filter *C* = [w1  w2]. and the derived state change computed.

III. EXPERIMENTAL SETUP AND RESULTS

*3.1 Experimental Setup*

In order to develop a combined method utilizing both types of motion estimation, the estimation error is determined by creating a constant movement around the desired axis and recording the error in the motion prediction for that Angular velocity ω . In order to test the accuracy and capability of the prediction algorithm, a special mount was developed for the phone system to standardize the angular rotation that the phone experiences. To achieve this a standard servo motor (HiTec HS645MG) was attached to an adjustable phone mount. The vertical angle of rotation can be adjusted offline while the lateral rotation is controlled by the attached servo to generate continuous motion, as seen in Fig. 5. The Botboarduino controlling the servo mount receives angular commands through the USB serial cable connected to the PC. The angular velocity ω of the mount is varied by writing a delay into the Botboarduino’s command script between each angular increment. The angular velocity is calculated by a different command script which returns the time duration of a particular motion, so that the motion can be divided by said time duration. The designed experimental setup allows the angle of the servo motor to directly act as the angular input. In summary, the main tested parameters for the system are the prediction accuracy concerning the angle amplitude and the angular velocity ω. No motion occurs in the image space (stationary environment), to restrict the mean angular change calculation to the environment itself. Because the vertical rotation of the camera phone is easy to estimate from the gravity vector, this work focuses on recreating the more difficult yaw rotation, the axis of which is parallel to the gravitational direction.

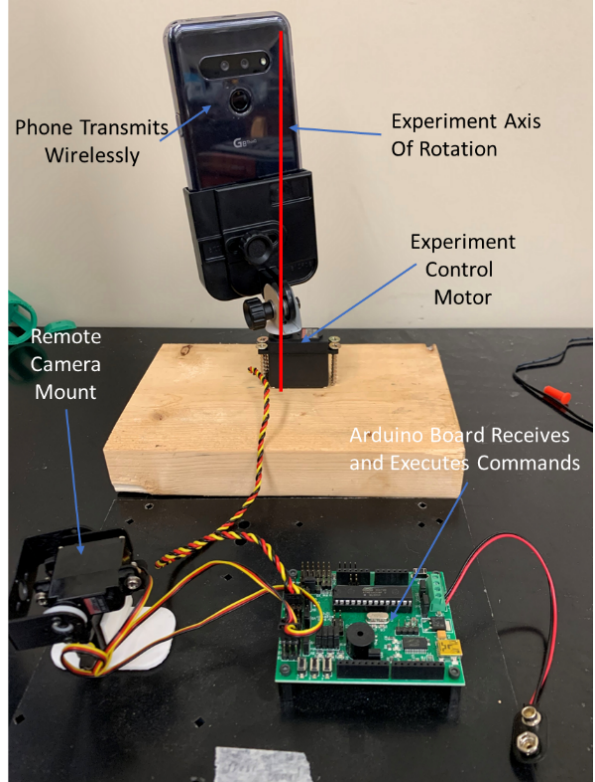


Fig. 5 Experimental Setup.

*3.2 Experiment Design*

For the experiment, the motion path was held constant while the relative angular velocity was changed between experiments. The motion path of the experimental setup was defined as 60 degrees in the anticlockwise direction as viewed from above, a pause of one second, then motion of 120 degrees clockwise, (or to 60 degrees clockwise from the original position) with a one second pause in this position, and finally back to the starting position. A graphical representation of this motion can be deduced from Fig. 6, where the predicted angular rotation very closely follows the motion path with very little error. The angular velocities of the phone mount were varied between 7.00. 7.63, 8.41, 9.63, 10.53, 12.03, 14.11, 15.35, 16.41, 18.87, 21.18, 25.57, 28.48, 34.35, and 43.35 degrees/s. For continuity, evaluation was performed on the sensor-based estimation technique at all angular velocities, even on the slower rates of motion where the estimation error is expected to be large. Similarly, evaluation was performed on the image-based technique, but above 21.18 deg/s the estimation fails completely, as the fast motion and narrow field of view create issues in overlap of images, and keypoints cannot be correctly matched. The error rate of each angular velocity trial is taken from the error in the estimation of the control signal, and each trial is repeated three times, with the resultant error taken together to compute the standard deviation.

*3.3 Results and Discussion*

In Figs. 6 and 7 the best illustration of the two separate methods is shown to highlight the range at which each of the two separate methods performs well in isolation. The angular velocity ω of the particular trial is included in the title. It can be seen in Fig. 7 that the larger motion rate of the faster trials well supports the sensor based approach, where the standard deviation was only 4.789 degrees at the highest velocity, very low for a numerical integration approach to noisy data, and almost twice the speed that the image based approach has failed completely. This type of motion would correlate to a quick user intent to see to the side of the direct field of vision. In Fig. 6 the smooth continuous motion of the image based approach is displayed, where slower motion can be well approximated by the keypoint mean shift, and the average error is only 1.938 degrees in the slowest case. This type of motion can be seen a user intent to slowly sweep to surroundings, or to track a slow-moving object in frame. Of note is the difference in the ramp of the two different motions, since the discrete state change calculations are very smooth in either case, giving the appearance of continuity.

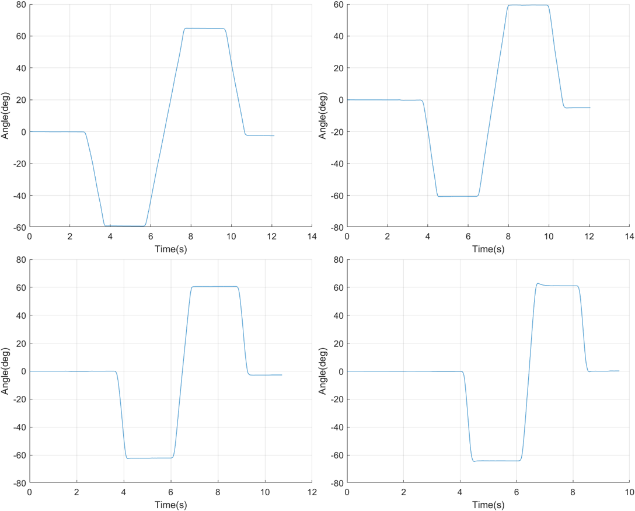


Fig. 6 Sensor ω =25.57, 24.48, 34.35, 43.35 deg/s

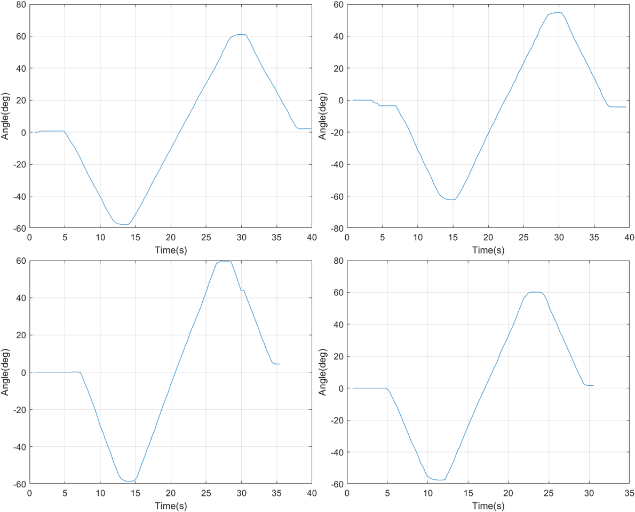


Fig. 7 Image ω = 7.00, 7.63, 8.41, 9.33 deg/s

More important than the performance of each at any particular instance is the expected standard deviation (std) of the degree error as a function of the angular velocity. The error is expected to be increasing exponentially for the image-based approach as distortion and keypoint matching difficulty become more prominent. Error is expected to decrease as a negative exponential function with increased angular velocity, which is a function of the signal-to-noise ratio (SNR). This trend should remain stable until the motion duration begins to approach the sampling frequency, which did not occur in any of the experimental trials. The standard deviations are given in Table 1 here, as well as the relative SNR to the control signal.

Table 1. Measured Standard Deviations



Figure 8 displays the fitted functions of error for the two approaches, with the red showing the sensor-based data, and the blue displaying the image based approach. The two approximate equations of error from above are:

\documentclass{article}
\usepackage{amsmath, amsfonts, amssymb}
\pagestyle{empty}
\begin{document}

$\begin{array}{c}
err_{i}=1.255e^{0.113\omega}\\
err_{s}=31.662e^{-0.063\omega}
\end{array}\;\;\text{(6)}$

\end{document}

The Kalman filter derived from the above was employed on the five sets of data that fell into the effective range of the algorithm, which is from 10 deg/s to 20 deg/s. The results of the Kalman filter combined method are displayed in Table 2.

Table 2: Kalman Filter Standard Deviations



The results of the various experiments to determine the standard deviations of each method individually and the combined Kalman filter method are depicted in here,

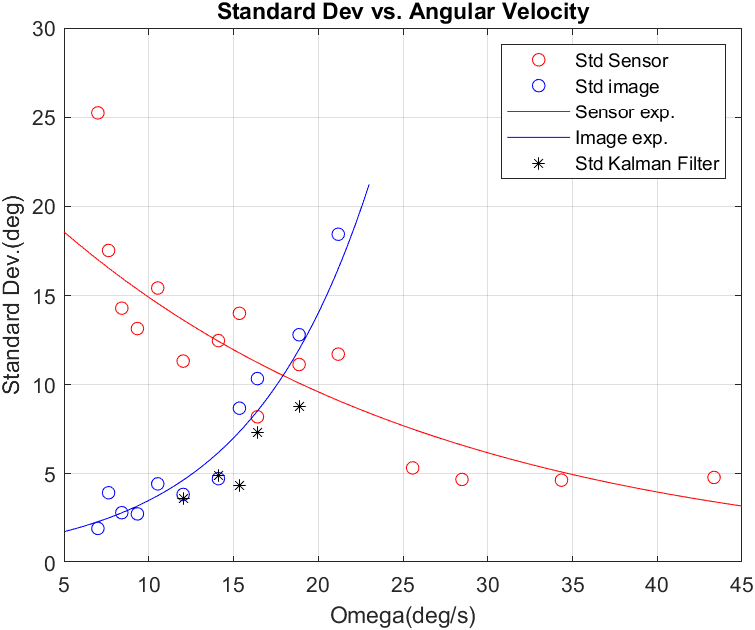


Fig. 8 Standard Dev as a Function of Angular Velocity

of particular note, the Kalman filter estimation performs better than either the sensor estimation of the image estimation for all but one of the angular velocities defined in the application range. The final angular velocity estimated in the range of the Kalman filter, 18.87 deg/s performed far better than either the image-based technique or the sensor based technique, with a 23 percent difference between the standard deviation of the two.

IV. Conclusion and future Work

In this work, a system based on a remote communication scheme meant to continuously estimate user motion that mimics how humans intuitively use a smartphone camera was developed and the accuracy of the system in predicting angular motion was tested, combining both sensor-based and image based user motion estimation, each of which produced a different error as a function of angular velocity. The sensor-based approach was taken to be exponentially decreasing, which correlates to the higher SNR. The image-based approach is taken to be exponentially increasing, which correlates to increasing distance approximation error and increasing matching error. By applying a Kalman filter approach where both methods tend to perform poorly, the standard deviation of the measurement error was significantly decreased compared to simply using either method, and overall the developed system accurately estimates angular change at a variety of different rotational velocities.

In future iterations of the system design, a higher communication speed through adjustment of the network communication scheme will be implemented to aid in increasing the accuracy and speed of the image-based approach to angular rotation prediction. The estimation system is to be implemented in real time on the system, such that the camera mount moves according to the state estimation achieved in this work. Finally, the remote camera mount component will hold a wireless video camera, which will transmit its image feed back to the user interacting with the camera phone, completing the full feedback loop of the intuitive remote camera mount system.

References

1. Rubio, F., Llopis-Albert, C., Valero, F. and Suñer, J.L, 2016, “Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory”. Robotics and Autonomous Systems, 86, pp.106-112.
2. Slamani, M., Nubiola, A. and Bonev, I., 2012. “Assessment of the positioning performance of an industrial robot”. Industrial Robot: An International Journal.
3. Rubio, F., Valero, F., Sunyer, J. and Mata, V., 2009. “Direct step‐by‐step method for industrial robot path planning”. Industrial Robot: An International Journal.
4. Cakmak, M. and Takayama, L., 2013, March. “Towards a comprehensive chore list for domestic robots.” In 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI) (pp. 93-94). IEEE.
5. Prassler, E., Munich, M.E., Pirjanian, P. and Kosuge, K., 2016. *Domestic robotics. In Springer handbook of* robotics (pp. 1729-1758). Springer, Cham.
6. Palacin, J., Salse, J.A., Valgañón, I. and Clua, X., 2004. “Building a mobile robot for a floor-cleaning operation in domestic environments.” IEEE Transactions on instrumentation and measurement, 53(5), pp.1418-1424.
7. Sutter, C., Oehl, M. and Armbrüster, C., 2011. “Practice and carryover effects when using small interaction devices”. Applied ergonomics, 42(3), pp.437-444.
8. Liu, H. and Wang, L., 2018. “Gesture recognition for human-robot collaboration: A review”. International Journal of Industrial Ergonomics, 68, pp.355-367.
9. Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A. and Mulanda, D., 2004. “Humanoid robots as cooperative partners for people”. Int. Journal of Humanoid Robots, 1(2), pp.1-34.
10. Kaufman, R.E., 2007, March. “A family of new ergonomic harness mechanisms for full-body natural constrained motions in virtual environments”. In 2007 IEEE Symposium on 3D User Interfaces. IEEE.
11. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L. and Di Marca, S., 2006. “Accelerometer-based gesture control for a design environment”. Personal and Ubiquitous Computing, 10(5), pp.285-299.
12. Luinge, H.J., Veltink, P.H. and Baten, C.T., 1999, October. “Estimation of orientation with gyroscopes and accelerometers”. In Proceedings of the First Joint BMES/EMBS Conference. 1999 IEEE Engineering in Medicine and Biology 21st Annual Conference and the 1999 Annual Fall Meeting of the Biomedical Engineering Society (Cat. N (Vol. 2, pp. 844-vol). IEEE.
13. Favre, J., Jolles, B.M., Siegrist, O. and Aminian, K., 2006. “Quaternion-based fusion of gyroscopes and accelerometers to improve 3D angle measurement”. Electronics Letters, 42(11), pp.612-614.
14. Zhang, X., Chen, X., Wang, W.H., Yang, J.H., Lantz, V. and Wang, K.Q., 2009, February. „Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors”. In Proceedings of the 14th international conference on Intelligent user interfaces (pp. 401-406).
15. DelPreto, J. and Rus, D., 2020, March. “Plug-and-Play Gesture Control Using Muscle and Motion Sensors”. In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (pp. 439-448).
16. Benko, H., Saponas, T.S., Morris, D. and Tan, D., 2009, November. “Enhancing input on and above the interactive surface with muscle sensing”. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (pp. 93-100).
17. Liu, Y.H., Huang, H.P. and Weng, C.H., 2007. “Recognition of electromyographic signals using cascaded kernel learning machine”. IEEE/ASME Transactions on Mechatronics, 12(3), pp.253-264.
18. Mian, A.S., Bennamoun, M. and Owens, R., 2008. “Keypoint detection and local feature matching for textured 3D face recognition”. International Journal of Computer Vision, 79(1), pp.1-12.
19. Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. “Object detection with deep learning: A review”. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.
20. Lowe, D.G., 2004. “Distinctive image features from scale-invariant keypoints”. International journal of computer vision, 60(2), pp.91-110.
21. Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L., 2008. “Speeded-up robust features (SURF)”. Computer vision and image understanding, 110(3), pp.346-359.
22. Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., 2011, November. “ORB: An efficient alternative to SIFT or SURF”. In 2011 International conference on computer vision (pp. 2564-2571). IEEE.
23. Alatise, M.B.; Hancke, G.P. Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. Sensors 2017, 17, 2164.
24. S. I. Roumeliotis and G. A. Bekey, "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 2958-2965 vol.3