2016 Indian Control Conference (ICC)
Indian Institute of Technology Hyderabad
January 4-6, 2016. Hyderabad, India

Interactive Mobile Interface with Augmented Reality for Learning Digital Control Concepts

Jared A. Frank[1], Anthony Brill[1], and Vikram Kapila[1]

Abstract— **The use of augmented reality (AR) and mobile applications has recently been investigated in the teaching of advanced concepts and training of skills in a variety of fields. By developing educational mobile applications that incorporate augmented reality, unique interactive learning experiences can be provided to learners on their personal smartphones and tablet computers. This paper presents the development of an immersive user interface on a tablet device that can be used by engineering students to interact with a motor test-bed as they examine the effects of discrete-time pole locations on the closed-loop dynamic response of the test-bed. Specifically, users point the rear-facing camera of the tablet at the test-bed on which colored markers are affixed to enable an image processing routine running on the tablet to measure the angular position of an arm attached to the motor. To perform vision-based control of the angular position of motor arm, a discrete-time Kalman filter and a full-state feedback controller are implemented in the background of the application. As the user taps on the touchscreen of the device, s/he adjusts the angular position of a 3D semi-transparent virtual arm that represents the set point to the system. An interactive pole-zero plot allows users to tap at any desired location for the closed-loop pole-placement, in turn triggering the application code to redesign a new controller for driving the test-bed. Real-time plots enable the user to explore the resulting closed-loop response of the test-bed. Experimental results show several responses of the test-bed to demonstrate the efficacy of the proposed system.**

I. Introduction

Augmented reality, or AR, is the overlaying of virtual content onto the real worldview. By implementing this state-of-the-art technique with hardware such as head mounted displays, projected displays, mobile devices, and even backpacks [22], mobile AR systems have been developed that provide informational support to users to aid them in accomplishing a variety of tasks [11]. Performing difficult tasks while following complicated instructions gives rise to the alternating attention problem that can impose significant cognitive demands on the user [26]. The benefit of AR in allowing users to maintain attention on the task has proven to be particularly beneficial in helping them prepare for work in potentially hazardous environments that require significant real-world experience, such as assembly and construction [4], [23], maintenance and inspection [9], surgery [8], and military operations [28].

Both image-based and location-based AR experiences have been investigated in science education [3]. For young students, these techniques have been shown to be useful in building knowledge and skills that rely on spatial relations, such as gaining an understanding in earth-sun relationships like rotation and revolution, solstice and equinox, and seasonal variation of light and temperature in the hemispheres [15], [25]. In higher education, as theoretical concepts and technological tools become more complex, students are in an increased need of innovative learning techniques. To meet these needs, AR has been incorporated in the teaching of 3D geometry in engineering graphics and mechanics classes [14], [20], the 3D structure of molecules in chemistry classes [19], and the arrangement of machine components in mechanical engineering classes [17]. To promote the use of portable augmented reality without the need for hardware such as projected or head-mounted displays, the implementation of AR has been examined on mobile devices [12], [18]. In recent years, interactive applications have been developed on mobile devices to teach everything from sign language [6] and astronomy [29], to more technical concepts like fourier series expansions [24].

In engineering courses and laboratories, the effects that the values of certain parameters have on the system of interest are not always well understood. Usually, curricula and textbooks take the traditional approach of providing few illustrative plots which serve as “snapshots” of the system at key demonstrative values of the parameters. However, this approach may overwhelm the students with multiple figures that they struggle to memorize without gaining a true understanding. An effective approach to examine the effects of variables on an engineering system utilizes experiential and situated learning with laboratory equipment. In engineering laboratories, user interfaces have evolved to incorporate virtual content and augmented reality to enhance the use of equipment such as atomic force microscopes [30] and to enhance the experience of remotely conducting automatic control experiments [1], [7]. The development of a mobile application with touch-based interaction and AR feedback for use in an automatic control laboratory can enhance the learning of non-intuitive concepts in digital control, such as the effect of closed-loop discrete-time pole locations on the dynamic system response.

To help students of automatic control achieve an understanding of the effects of discrete-time pole locations, an immersive interface is developed on a tablet device to control a motor test-bed in the laboratory. By overlaying a manipulable virtual arm on a live view of the motor test-bed, as it is captured by the rear-facing camera on the tablet, students are afforded an intuitive mechanism to command the angular

This work is supported in part by the National Science Foundation awards RET Site EEC-1132482, GK-12 Fellows DGE: 0741714, and DRK-12 DRL: 1417769, and NY Space Grant Consortium grant 48240-7887.

[1]Mechatronics and Control Lab, Mechanical and Aerospace Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201 [jared.alan, brill.anthony, vkapila]@nyu.edu

978-1-4673-7993-9/16/$31.00 ©2016 IEEE

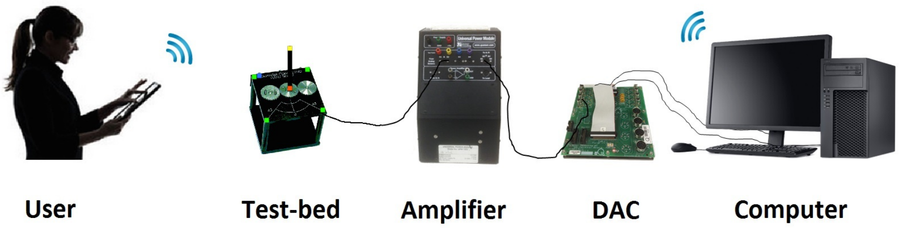


Fig. 1: Schematic representation of the system for vision-based closed-loop feedback control.

position of the actual arm of the motor as well as a visual aid to observe phenomena such as settling time, steady-state error, and oscillations about the set point. Interactive views, such as the virtual arm, real-time plots of the state and control responses, and an interactive pole-zero plot, are provided to engage students in the learning experience and allow them to conduct inquiry-based investigations with physical equipment from their personal mobile devices. By tapping at a location on the pole-zero plot, the student chooses the desired closed-loop poles of the system and examines the resulting experimental behavior.

The paper is organized as follows. In Section II, an overview of the system and user interface are presented. Section III describes the image processing routine executed on the mobile application to obtain vision-based measurements of the angular position of the motor arm. Section IV gives an overview of the application architecture while Section V discusses the sampled-data model of the motor test-bed used for the design and implementation of a Kalman filter for state estimation and a full-state feedback controller to drive the test-bed. Section VI presents results of the experiment conducted with the system to evaluate the performance of the wireless networked control system while Section VII offers some concluding remarks.

II. System Description

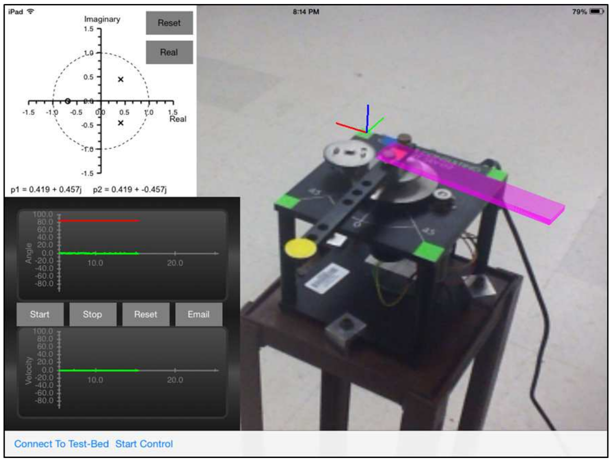
The system used in this study is composed of an educational motor test-bed with a 6 inch (.1524 meter) long rectangular metal arm. The motor is driven by a power amplifier which receives control signals from a desktop computer via a data acquisition and control (DAC) board. The desktop computer, which runs the MATLAB/Simulink program, transmits to the test-bed control signals that it receives over Wi-Fi from a mobile application executing on the tablet device held by the user. This mobile application captures video from the rear-facing onboard camera as it is pointed at the experiment from an arbitrary perspective. Colored markers affixed to the test-bed are detected by an image processing program running on the tablet application, and used to obtain vision-based measurements of the angular position of the motor arm. The measured position is fed into recursive state estimation and feedback control algorithms to compute the control action that is wirelessly sent to the desktop computer. The components that comprise the system are shown in Figure 1.

A. Smart Device

The smart mobile device used in this study is an Apple iPad 2, which has a 9.7 inch (250 mm), 1024×768 pixel multi-touch display, 1 GHz dual-core processor, and a 0.7-megapixel rear-facing camera. Apple software development supports the use of open source and third-party libraries, such as Open Source Computer Vision Library (OpenCV), Open Source Graphics Library (OpenGL), and CocoaAsyncSocket for image processing, rendering of augmented reality content, and TCP/IP communication, respectively.

B. User Interface

The user interface developed for this work (see Figure 2) is split into three main views. In a large view on the right, a live video stream from the rear-facing camera on board the tablet is shown at 30 frames per second. Projected onto this view is a purple virtual arm, which lies in the horizontal plane with the actual motor arm in the video and represents the set point for the system. One end of the virtual arm is fixed to the orange marker attached to the corresponding end of the actual motor arm, and the other end pivots about this point - in the same manner that the actual motor arm rotates about its axis - to face a location on the screen that is tapped by the user. Although the touch on the touchscreen is 2-D in nature, it is able to naturally manipulate the 3-D orientation of the virtual arm. Section III describes the marker-based image processing approach behind this interaction as well as the rendering of the augmented reality content in the real world view.



p1 = 0.419 + 0.457j   p2 = 0.419 + -0.457j

Fig. 2: Screenshot from the user interface of the application.

The two left-hand views of the interface contain interactive plots: on top users tap on the screen to select desired closed-loop poles of the system and on bottom the experimental response of the system is displayed. In the top left view, the discrete-time pole-zero plot uses a touch gesture recognition class to detect taps on the screen, whose locations are mapped from screen coordinates to coordinates on the plot. Based on the system parameters, the zero of the closed-loop system is automatically determined and placed at the appropriate location on the plot. Two buttons are available that allow the user to reset the plot or to switch between two modes of pole selection. In the first mode, only the x coordinate of the tapped location is used to place a pole on the real axis of the plot. In the second mode, both the x and y coordinates are used to produce a complex pole at the tapped location as well as at the location of the corresponding complex conjugate pole. This triggers a pole-placement formula to be run by the application to compute the corresponding control gains to be used to drive the test-bed (see Section V). The result is an interface that allows students to interactively explore the effects of discrete-time pole locations on the system performance by viewing the motor response and comparing the angular position of the motor arm to that of the virtual arm, looking over the tablet device to watch the actual arm's motion, or by observing a plot of the angular position response as it is generated in the bottom left view. Buttons in the plot view enable the user to start, stop, and reset the plots at any time as well as email collected data for post-processing.

III. Computer Vision

The interface used in the proposed system allows students to observe the motor test-bed from an arbitrary perspective (provided all seven colored markers at the top of the test-bed are in view), and still collect accurate vision-based measurements of the angular position of the motor arm for feedback control of the test-bed. Moreover, the application allows users to tap at a location in the 2-dimensional view and maps this location on the screen to the location in the plane of the motor arm for rendering the virtual arm such that it is oriented toward that location in the plane. To provide these abilities, the application exploits the use of colored markers placed on the test-bed in two parallel planes.

A. Homography

A reference image of the test-bed, with the image plane oriented approximately parallel to the two planes of the test-bed containing the markers, is stored by the application (see Figure 3). The coordinate frame of this image is used as the reference frame for the remainder of the experiment. The locations of the centers of four green markers can then be obtained in this reference frame and matched with the locations of the same four green markers in the coordinate frame of each frame captured by the onboard camera as it is pointed at the test-bed from an arbitrary perspective while the experiment is running. Since these markers lie on a 2-D plane in 3-D space, a projective homography matrix, G, can be calculated [10]

$$\alpha_g p = G p^*.$$

This matrix defines the transformation between a point $p = (u, v, 1)$ in pixels in an arbitrary coordinate system and the point $p^* = (u^*, v^*, 1)$ in the reference coordinate system, up to a scale factor α_g.

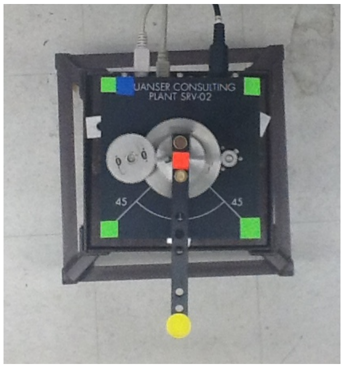


Fig. 3: Reference image used to calculate the homographies associated with arbitrary perspectives.

B. Marker Detection

To detect the markers on each of the two planes of the test-bed, a color segmentation approach is taken. The approach involves first empirically determining separate color ranges in the hue-saturation-value (HSV) space for each colored marker, and then thresholding the image according to each of the established ranges to yield binary images whose white pixels represent pixels belonging to detected markers. To remove small amounts of noise, morphological open and close operations are performed on the binary images at the expense of some computational cost [27]. Center coordinates of the markers are obtained by first calculating the contours that border each region of filtered white pixels in each binary image and then using the concept of area moments to compute centers of these regions.

Experiments reveal that over 90% of the computation required by the application is involved in the detection of the colored markers on the test-bed. A technique is used that significantly reduces the search space for markers at each time step. In each frame, a square region of interest is created that is centered on the previously calculated center point for each marker. This region of interest is obtained by calculating the minimal up-right bounding rectangle around the marker from the array of white pixels in the thresholded binary image, and then scaling up the size of the rectangle by a a factor of 2. Next, in the following frame, the search for each marker is performed within each of these small windows in the neighborhood of the marker's previously determined location. A similar technique has been shown to speed up the process of lane detection in vision-based traffic applications [13]. In the proposed system, the computational load is reduced such that a frame rate of 30 Hz can be easily maintained.

On the first plane, which lies on the surface of the base of the test-bed, one blue marker and four green markers are strategically placed. Each of the green markers is positioned at a separate corner of the base, forming the four corners of a square. These markers, whose center coordinates are stored in a vector, may not be detected in the same order from one frame to another frame or from one perspective to another perspective. Therefore, to keep the order of the markers consistent in each frame and from any arbitrary perspective, so that a homography can be reliably calculated, a blue marker is introduced whose center coordinates serve as a reference location with which the distances to the centers of the green markers may be compared, as shown in Figure 4. By placing the blue marker between two corners on an edge of the square, but closer to one corner than the other, a configuration is achieved in which the four green markers can be uniquely identified from most reasonable downward-facing perspectives of the tablet, after they are sorted in the order of increasing distance from the center of the blue marker in units of pixels.

After the blue marker has been used to sort the green markers such that they can be matched from each frame of the video capture and the reference image, and the green markers have been used to calculate the homography between the two image frames, the orange and yellow markers placed on the motor arm in a plane above but parallel to the plane of the green and blue markers must be detected. These markers are used to calculate the angular position of the motor arm rotating around the vertical axis of the reference image.

C. Motor Angle Measurements

To establish a reference orientation with which to calculate the angular position of the motor arm, the vertical direction of the reference frame is used. After the image processing routine has used the color segmentation approach to detect the green markers and compute the homography matrix associated with the frame at the current time step, the coordinates of the center points of the orange and yellow markers are detected with respect to the current frame. These coordinates are then transformed back into the reference coordinate system by way of the inverse of the homography matrix.

Note that the orange and yellow markers are not located in the same plane as the blue and green markers on the surface of the base of the test-bed. Therefore, in general, this is a violation of the prerequisite condition for obtaining reliable 2D projected locations of the markers. For example, consider an arbitrary view of the test-bed, as shown in Figure 5(a). After computing the homography matrix associated with this perspective, taking its inverse, and using the resulting matrix to transform the pixels of this view into the reference frame, we obtain the view shown in Figure 5(b). As can be examined by observation, the spatial relations between pixels lying in the plane of the green and blue markers seems in proportion to the relations in the reference image in Figure 3, while the spatial relations associated with pixels that make up objects not in this plane are clearly skewed with respect to those pixels. For instance, we know that the motor arm and leftmost gear are not actually so far left of the center of the base, and that the top and bottom of the test-bed stand are not as far right of the center of the base. However, note that the spatial relations between the points that describe the motor arm and its markers, between the points that describe the top of the test-bed stand, and between the points that describe the bottom of the test-bed stand all appear in proportion with respect to each other. This reveals an interesting observation: although the transformation of the arbitrary view distorts the spatial relations between points that belong to different planes parallel to the reference image plane, and distort by different amounts depending on the distance each plane is from the reference image plane, they nevertheless distort points belonging to the same plane equally. Therefore, angular relationships between points belonging to the same plane remain unchanged, and since this application of homography is required to determine only the angular orientation of the motor arm, it is still capable of obtaining accurate angular measurements of the motor arm.

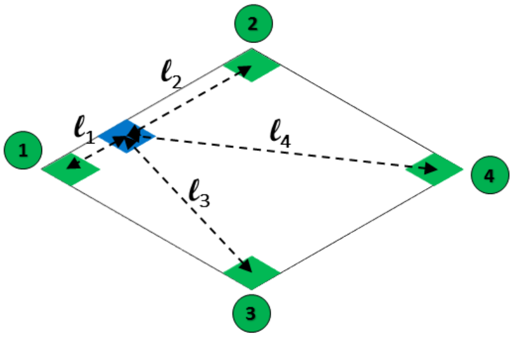


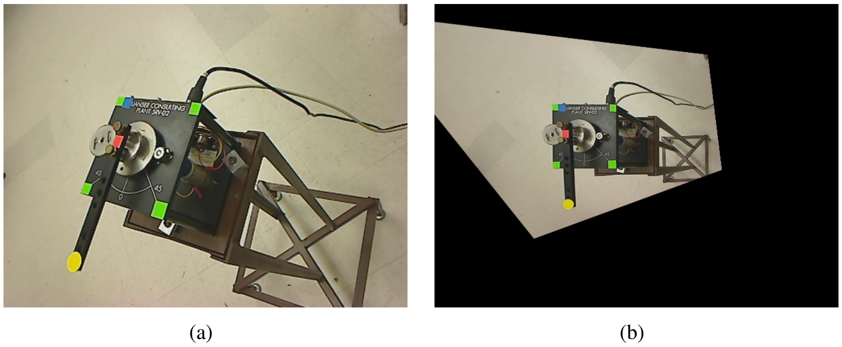
Fig. 4: Diagram of the markers placed on the surface of the base of the motor test-bed such that $l_1 < l_2 < l_3 < l_4$ for most practical perspectives.

Once the location of the orange and yellow markers have been converted into the reference frame, the angle of the motor arm in radians can be determined. The coordinates of the orange marker $p_o^* = (x_o^*, y_o^*)$ and yellow marker $p_y^* = (x_y^*, y_y^*)$ are used in the inverse tangent function as follows

$$\theta = -\tan^{-1}\left(\frac{y_o^* - y_y^*}{x_o^* - x_y^*}\right).$$

D. Camera Pose for Rendering Augmented Content

The rendering of augmented reality content in the scene is critical for achieving sensory immersion and facilitating the user's suspension of disbelief in the touch-based interaction with the content. To overlay the 3D virtual arm onto frames as they are captured by the camera, the pose of the arm must be known relative to the coordinate frame of the camera. This pose is a homogeneous transformation composed of a rotation matrix and translation vector in Euclidean coordinates. Rendering the virtual arm such that it is registered in the perspective view of the test-bed (i.e., such that it appears in the plane with the motor arm) requires the construction of a 3D coordinate system in the image with its origin at the



(a)

(b)

Fig. 5: Images of the test-bed obtained (a) from an arbitrary perspective and (b) after warping the image displayed in (a) using the inverse of the homography matrix calculated for the perspective depicted in (a).

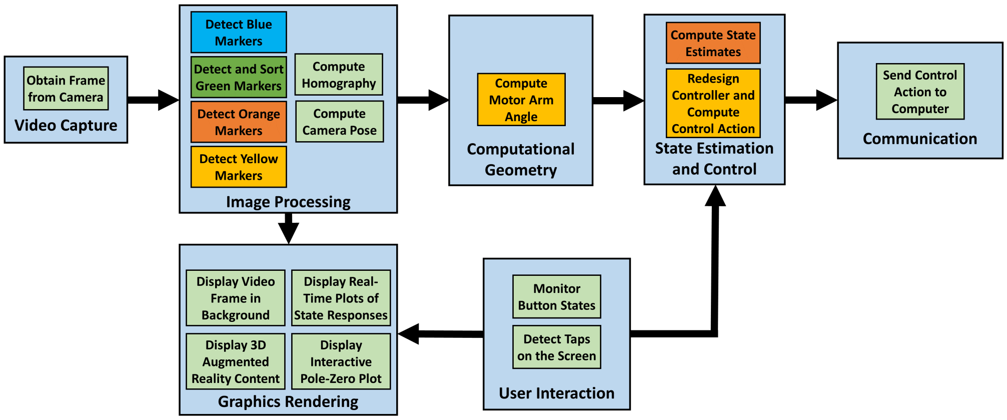

Obtain Frame from Camera
Video Capture
Detect Blue Markers
Detect and Sort Green Markers
Compute Homography
Detect Orange Markers
Compute Camera Pose
Detect Yellow Markers
Image Processing
Compute Motor Arm Angle
Computational Geometry
Compute State Estimates
Redesign Controller and Compute Control Action
State Estimation and Control
Send Control Action to Computer
Communication
Display Video Frame in Background
Display Real-Time Plots of State Responses
Display 3D Augmented Reality Content
Display Interactive Pole-Zero Plot
Graphics Rendering
Monitor Button States
Detect Taps on the Screen
User Interaction

Fig. 6: Flowchart illustrating the architecture of the application developed for this study.

center of the first green marker. By expressing the locations of the remaining three markers in this coordinate system, calibrating the device camera (i.e., obtaining the values of intrinsic parameters of the camera such as focal lengths, pixel coordinates of principal point, and lens distortion parameters [31]), and obtaining the projected locations of the green markers using the aforementioned color segmentation approach, the pose of the camera can be estimated with respect to the 3D coordinate system. This pose is found by solving a correspondence problem between the 2D projected points and the 3D points [2]. However, to render the 3D content, this camera pose must be inverted to give the pose of the 3D coordinate system with respect to the camera frame.

IV. Application Architecture

Before the recursive Kalman filter and angular position control algorithms are implemented on the mobile platform, a Simulink model is designed on the desktop computer and used to control the motor system at the proposed sampling rate (30 Hz). The MATLAB/Simulink environment is also used to design the Kalman gain and feedback control gain and test the designs in the closed-loop control of the test-bed. On the mobile platform, these algorithms are easily implemented in an object-oriented program, since data structures provided by OpenCV allow for the execution of the necessary 2nd-order recursive matrix-vector equations involved in the control design and in the estimation and control computations (see Section V). Figure 6 outlines the modular architecture implemented in the tablet application code using an object-oriented approach.

V. Modeling, Estimation, and Control

To control the angular position of the motor arm on the test-bed using vision-based measurements from the image processing routine on the tablet, a sampled-data model of the plant dynamics is obtained to support the design of a discrete-time Kalman filter for state estimation and pole-placement algorithm for control design (see Figure 7).

A. Model

The system used in this study consists of an armature controlled motor, gearbox, and motor arm. Its dynamic behavior is represented by a first-order transfer function from the input voltage $U(s)$ to the angular velocity of the motor arm $\Omega(s)$

$$\frac{\Omega(s)}{U(s)} = \frac{K}{\tau s + 1}.$$

In this model of the plant, K is the steady-state DC-gain and τ is the time constant of the system. Both parameters K

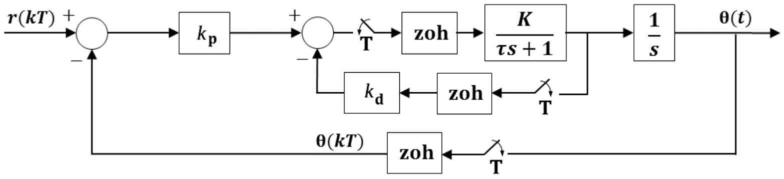


Fig. 7: Block diagram representing the sampled-data approach to modeling the wireless networked control of the motor.

and τ are determined experimentally by applying a unit-step input to the system

$$K = 1.5696V, \quad \tau = 0.036s.$$

A state-space representation for this system can be obtained in which the state $x(t)$ is composed of the angle $\theta(t)$ of the motor arm and its angular velocity $\omega(t)$ ($x(t) \triangleq [\theta(t) \quad \omega(t)]^{\mathrm{T}}$)

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{K}{\tau} \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t).\end{aligned}$$

Note that measurement $y(t)$ consists of just the angular position of the motor arm. Since this measurement is sampled at a fixed rate of 30 Hz (the frame rate of video being captured by the camera), the above continuous-time model is discretized at each sampling instant kT, $k = 0,1,2,\ldots$, where $T = \frac{1}{30}$ second is the sampling period

$$\begin{aligned}x[(k+1)T] &= \phi(T)x[kT] + \gamma(T)u[kT], \\ y[kT] &= Cx[kT].\end{aligned}$$

In this model, $\phi(T) \triangleq e^{AT}$ is the state transition matrix of the continuous-time state matrix A, and $\gamma(T) \triangleq \int_0^T \phi(T-\tau)Bd\tau$.

Before the Kalman filter and full-state feedback controller can be designed, the controllability and observability of this discretized model must be confirmed. This is done by assuring the full rank of the following controllability matrix $M_c(T)$ and observability matrix $M_o(T)$

$$\begin{aligned}M_c(T) &= \begin{bmatrix} \gamma(T) & \phi(T)\gamma(T) \end{bmatrix}, \\ M_o(T) &= \begin{bmatrix} C \\ C\phi(T) \end{bmatrix}.\end{aligned}$$

B. State Estimation

Although the plant model is second-order, the image processing routine is designed only to obtain measurements of the angle $\theta(kT)$. Therefore, only one of the states needed for full-state feedback control is being measured. Moreover, the vision-based angular position measurement will be corrupted by noise due to imperfect image quality and variations in scene illumination. To obtain filtered estimates of the full state of the discretized model, a Kalman filter is implemented at each time step k by recursively executing the following equation

$$\begin{aligned}\hat{x}[(k+1)T] = \phi(T)\hat{x}[kT] &+ \gamma(T)u[kT] \\ &+ L(y[kT] - C\hat{x}[kT]).\end{aligned}$$

In this equation, $L = (\phi QC^{\mathrm{T}})(CQC^{\mathrm{T}} + V_2)^{-1}$ is the Kalman gain matrix, where V_2 is the noise covariance matrix of the measurements and Q is the matrix solution of the discrete-time algebraic Riccati equation [16]

$$Q = \phi Q \phi^{\mathrm{T}} - (\phi QC^{\mathrm{T}})(V_2 + CQC^{\mathrm{T}})^{-1}(\phi QC^{\mathrm{T}})^{\mathrm{T}} + V_1.$$

In this Riccati equation, V_1 is the noise covariance matrix of the process. Although this process noise is not measured in a straightforward manner, the measurement covariance V_2 can be obtained by studying measurement data collected over a sufficient time period. Then, the process noise is tuned to achieve a relatively high level of confidence in the model with respect to the measurements. This is done by assigning a process noise covariance matrix of $V_1 = 10^{-6} \times I_4$, a diagonal matrix with elements very small in comparison to the elements of the experimentally determined measurement noise covariance $V_2 = \mathrm{diag}(0.0024^2, 0.004^2)$.

C. Control Design

To design the full-state feedback controller $K = [k_{\mathrm{p}} \quad k_{\mathrm{d}}]$ such that the closed-loop system has the desired poles p_1, p_2, the desired characteristic equation is determined as follows

$$\lambda(z, p_1, p_2) = (z - p_1)(z - p_2) = z^2 - (p_1 + p_2)z + p_1 p_2.$$

Ackermann's formula is used to compute the controller K from the controllability matrix M_c and the matrix obtained by plugging the state matrix ϕ into the desired characteristic equation [5]

$$K(p_1, p_2) = [0 \quad 1]M_c^{-1}\lambda(\phi, p_1, p_2),$$

where

$$\lambda(\phi, p_1, p_2) = \phi^2 - (p_1 + p_2)\phi + (p_1 p_2)I_2.$$

Having determined the control gains, now the control signal is computed as follows

$$u(kT) = K\hat{x}(kT), \quad k = 0,1,2,\ldots,$$

using the state estimate $\hat{x}(kT)$.

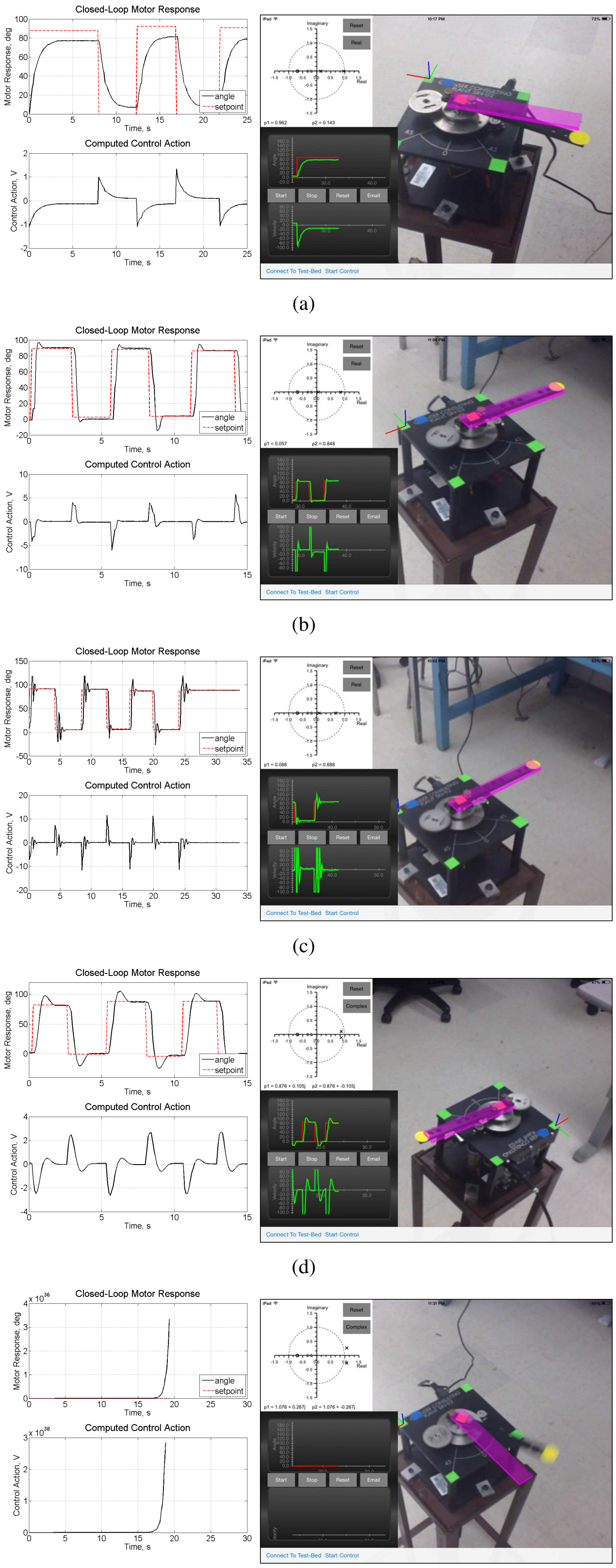
VI. Experimental Results

To investigate the potential of the proposed application to perform vision-based control of the motor test-bed using a full-state feedback controller that is redesigned when the user taps at desired closed-loop pole locations, an experiment is performed in which the tablet is held facing the test-bed from various perspectives. The response of the system is recorded and plotted on the application in real-time as a user taps on the screen to issue step commands to the motor at approximately 0° and 90°. These data from the interactive plots are saved and emailed, and then imported into MATLAB for post-processing. Figure 8 shows the responses of the motor test-bed plotted with the setpoints issued at several key locations of the closed-loop poles, as well as plots of the computed control action displayed underneath. Screenshots from the tablet application are displayed next to the plots while the application is being used to collect data and control the test-bed.

As shown in the pole-zero plot in the Figure 8(a), the first poles selected by the user are on the real axis at $p_1 = 0.962, p_2 = 0.143$, with one pole near to the origin dominated by the other pole close to the $(1,0)$ point. Therefore, the system responds relatively slowly to step commands, with an average settling time of 2.8 seconds. Because the motor exhibits deadzone behavior and therefore doesn't respond to control signals with magnitude less than approximately 0.2 Volt, there is an average steady-state error of approximately 10.86°. Similar to the first pole configuration, the second pole configuration, $p_1 = 0.057, p_2 = 0.848$, causes the system to behave as a first-order system with the dominant pole shifted closer toward the origin (see Figure 8(b)). The result is a significantly faster response, with an average settling time of 0.667 second. In theory, this step response should not exhibit any overshoot, however due to effects such as small time delays in the wireless communication and unmodeled mechanical effects of the motor, there is a small overshoot of approximately 7.62%. As with the previous two pole configurations, the system with the following pole configuration, $p_1 = 0.086, p_2 = 0.686$, behaves as a first-order system. With the dominant pole shifted even closer toward the origin, the response now demanded by the controller is unable to be met by the system, which has limited bandwidth and ± 5 Volts actuator saturation (see corresponding control signal in Figure 8(c)). The result is a response with a very fast peak time, but with large overshoot (30.48%) and settling time (1.25 seconds) due to these destabilizing effects.

To explore oscillations as they are purposefully introduced by control design, complex conjugate poles are chosen at $p_1, p_2 = 0.876 \pm 0.105\imath$. This pole configuration results in a response with 22.16% maximum overshoot and a large settling time of 1.53 seconds (see Figure 8(d)). To demonstrate a discrete-time closed-loop pole configuration that makes the system unstable, a pair of complex poles are selected outside of the unit circle, at $p_1, p_2 = 1.076 \pm 0.267\imath$ (see Figure 8(e)). Table I lists the values of the controller gains computed to accomplish the digital control of the test-bed from the tablet.

In addition to obtaining experimental results using the tablet, simulations of the system are run in the MATLAB/Simulink environment. Although the simulated results show many similar trends in system performance as the experimental plots, values of the time-domain characteristics are not identical due to the unmodeled properties of communication delays and motor. Table II summarizes the experimental and simulation results of the analysis. A video of the experiment is available at [21].



(a)

(b)

(c)

(d)

(e)

Fig. 8: Closed-loop responses of the system (left) next to screenshots of the application controlling the test-bed (right).

TABLE I: Selected pole configurations and corresponding control gains.

Poles	Controller
0.143, 0.962	$K = [-0.8948 \quad -0.1763]$
0.057, 0.848	$K = [-3.9386 \quad -0.2992]$
0.086, 0.686	$K = [-7.8860 \quad -0.3429]$
$0.876 \pm 0.105\imath$	$K = [-0.7254 \quad 0.4130]$
$1.076 \pm 0.267\imath$	$K = [-2.117 \quad 0.8069]$

TABLE II: Performance characteristics of experimental and simulated responses of the motor test-bed to several digital full-state feedback controllers implemented on a tablet.

Poles	Experimental		Simulated	
	Overshoot (%)	Settling Time (s)	Overshoot (%)	Settling Time (s)
0.143, 0.962	N/A	2.8	N/A	2.433
0.057, 0.848	7.62	0.667	N/A	0.7667
0.086, 0.686	30.48	1.25	N/A	0.433
$0.876 \pm 0.105\imath$	22.16	1.53	2.92	1.067
$1.076 \pm 0.267\imath$	N/A	N/A	N/A	N/A

VII. Conclusion

This paper presented a mobile application that uses the rear-facing camera on board a tablet computer to perform vision-based control of a motor test-bed. Using touch-based interaction and augmented reality content, the user interface allows users to select desired discrete-time closed-loop poles, which the application uses to calculate full-state feedback control gains that are used to drive the system. The interface provides three separate interactive views (i.e., interactive pole-zero plot, interactive real-time state response and control signal plots, and live video of the experiment with interactive 3D motor arm) that users can use to reinforce connections and gain insights regarding the effect of the discrete-time pole locations. Future work will involve a comparative study to assess the educational outcomes of the proposed system in inquiry-based learning of digital control concepts in classrooms of undergraduate engineering students.

References

[1] J. M. Andujar, A. Mejías, and M. A. Marquez, “Augmented reality for the improvement of remote laboratories: an augmented remote laboratory,” *IEEE Transactions on Education*, vol. 54, no. 3, pp. 492–500, 2011.

[2] D. Baggio, *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing Ltd, 2012.

[3] K.-H. Cheng and C.-C. Tsai, “Affordances of augmented reality in science learning: suggestions for future research,” *Journal of Science Education and Technology*, vol. 22, no. 4, pp. 449–462, 2013.

[4] H.-L. Chi, S.-C. Kang, and X. Wang, “Research trends and opportunities of augmented reality applications in architecture, engineering, and construction,” *Automation in Construction*, vol. 33, pp. 116–122, 2013.

[5] R. Dorf and R. Bishop, *Modern Control Systems (11th ed.)*. New Jersey: Prentice Hall, 2007.

[6] S. A. El-Seoud *et al*, “A proposed pedagogical mobile application for learning sign language,” *Int. Journal of Interactive Mobile Technologies*, vol. 7, no. 1, pp. pp–46, 2013.

[7] J. Frank and V. Kapila, “Development of mobile interfaces to interact with automatic control experiments [focus on education],” *IEEE Control Systems*, vol. 34, no. 5, pp. 78–98, 2014.

[8] H. Fuchs *et al*, *Augmented reality visualization for laparoscopic surgery*. Springer, 1998.

[9] T. Haritos and N. D. Macchiarella, “A mobile application of augmented reality for aerospace maintenance training,” in *Digital Avionics Systems Conference*, vol. 1, 2005, pp. 5.B.3–1–5.B.3–9.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[11] T. Höllerer and S. Feiner, “Mobile augmented reality,” *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK*, vol. 21, 2004.

[12] W. Hürst and C. Van Wezel, “Multimodal interaction concepts for mobile augmented reality applications,” in *Advances in Multimedia Modeling*, 2011, pp. 157–167.

[13] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, “A survey of video processing techniques for traffic applications,” *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[14] H. Kaufmann and D. Schmalstieg, “Mathematics and geometry education with collaborative augmented reality,” *Computers & Graphics*, vol. 27, no. 3, pp. 339–345, 2003.

[15] L. Kerawalla *et al*, “”making it real”: Exploring the potential of augmented reality for teaching primary school science,” *Virtual Reality*, vol. 10, no. 3-4, pp. 163–174, 2006.

[16] F. Lewis, *Optimal Estimation: With an Introduction to Stochastic Control Theory*. Wiley New York et al., 1986.

[17] F. Liarokapis *et al*, “Web3d and augmented reality to support engineering education,” *World Transactions on Engineering and Technology Education*, vol. 3, no. 1, pp. 11–14, 2004.

[18] C. Liu *et al*, “Evaluating the benefits of real-time feedback in mobile augmented reality with hand-held devices,” in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 2973–2976.

[19] P. Maier, M. Tönnis, and G. Klinker, “Augmented reality for teaching spatial relations,” in *Conference of the International Journal of Arts & Sciences, Toronto*, 2009.

[20] J. Martín-Gutiérrez, M. Contero, and M. Alcañiz, “Evaluating the usability of an augmented reality based educational application,” in *Intelligent Tutoring Systems*, 2010, pp. 296–306.

[21] Mechatronics and Control Laboratory. (2015) Interactive mobile interface with augmented reality for learning digital control concepts. [Online]. Available: http://engineering.nyu.edu/mechatronics/videos/ardcmotor.html.

[22] W. Piekarski, R. Smith, and B. H. Thomas, “Designing backpacks for high fidelity mobile outdoor augmented reality,” in *Proc. IEEE/ACM Int. Symp. Mixed and Augmented Reality*, 2004, pp. 280–281.

[23] H. Regenbrecht, G. Baratoff, and W. Wilke, “Augmented reality projects in the automotive and aerospace industries,” *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 48–56, 2005.

[24] M. C. Reis *et al*, “Teaching fourier series expansions in undergraduate education with the help of the fouse android application,” *Int. Journal of Interactive Mobile Technologies*, vol. 8, no. 1, pp. pp–26, 2014.

[25] B. E. Shelton and N. R. Hedley, “Using augmented reality for teaching earth-sun relationships to undergraduate geography students,” in *IEEE Int. Workshop Augmented Reality Toolkit*, 2002, pp. 1–8.

[26] M. M. Sohlberg and C. A. Mateer, *Introduction to Cognitive Rehabilitation: Theory and Practice*. Guilford Press, 1989.

[27] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer-Verlag New York, Inc., 2003.

[28] C. Tappert *et al*, “Military applications of wearable computers and augmented reality,” *Fundamentals of wearable computers and augmented reality*, pp. 625–647, 2001.

[29] K. Tian *et al*, “Lunar observation support system using smartphone ar for astronomy education,” *Int. Journal of Interactive Mobile Technologies*, vol. 8, no. 1, pp. pp–32, 2014.

[30] W. Vogl, B. K.-L. Ma, and M. Sitti, “Augmented reality user interface for an atomic force microscope-based nanorobotic system,” *IEEE Transactions on Nanotechnology*, vol. 5, no. 4, pp. 397–406, 2006.

[31] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.