International Journal of Applied Engineering Research ISSN 0973-4562 Volume 16, Number 6 (2021) pp. 497-501

Stabilizing an Inverted Pendulum with a Virtual Video Sensor

Joseph Lupton, Brendon Ortolano, Josiah Keane, Michael Reynolds, PhD

Department of Mechanical Engineering, The University of West Florida, Pensacola, FL 32514, USA.

Abstract

This paper demonstrates how conventional video processing techniques can be implemented at a frequency sufficient to control an inverted pendulum. Further, it details the mechanical design of an inverted pendulum cart and demonstrates a unique application of virtual sensors far more challenging than their typical uses. The inverted pendulum is controlled with an LQR that utilizes image-based visual servoing techniques. We show that the computation time for video processing has been reduced to a level where fully on-board systems are feasible.

Keywords: Virtual Sensor, LQR, Inverted Pendulum, Controls, Visual Servoing

INTRODUCTION

The inverted pendulum (Figure 1) is a frequently attempted control problem that uses a cart to balance an upright pendulum. Historically, the inverted pendulum (IP) has been used a benchmark for controller designs as it is a non-linear, multivariable control problem. Most IP applications rely on the use of rotary encoders to determine the angle of the rod and, in certain cases, the position of the body. Both the angle and position are utilized in feedback control, meaning the entire state of the system is typically measured solely with encoders. In this application, we are replacing the pendulum encoder with a camera that reads the rod angle. With advancements in microprocessor speeds, it is now possible to balance an IP using a virtual video sensor. In this paper, we will demonstrate how we were able to balance the pendulum using visual servoing and give insight on the system design and performance.

Virtual (video) sensors have been used in a variety of control formats such as surveillance [1] , robotics [2], traffic control [3], and certain industrial processes [4]. With the increasing affordability and availability of computers and microprocessors, virtual sensors are becoming more feasible. Virtual sensors are sensing systems that use alternative information and measurements to calculate a value that may be measured directly by a sensor. Virtual sensors are finding niches in tasks that require sensing in areas where physical sensors cannot fit as well as tasks that require the use of AI. Typically, the application has been appropriate for the necessary time delay associated with video processing. Balancing an IP requires a very fast response with low time delay. One study on a similar model demonstrated that a time delay longer than 60 milliseconds can make stability nearly impossible [5]. Until recently, video processing with such a small time delay was not feasible on board an IP cart.

In this paper we will demonstrate how it is now possible to use a virtual sensor for challenging control applications such as the IP. We will first describe the hardware and software used in the system. Second, we will demonstrate that the sensor has sufficient bandwidth by comparing the response to a standard encoder. Third, we will show the results when applied on the IP system. It is hoped that this demonstration will show greater potential for virtual video sensors in a larger range of applications.

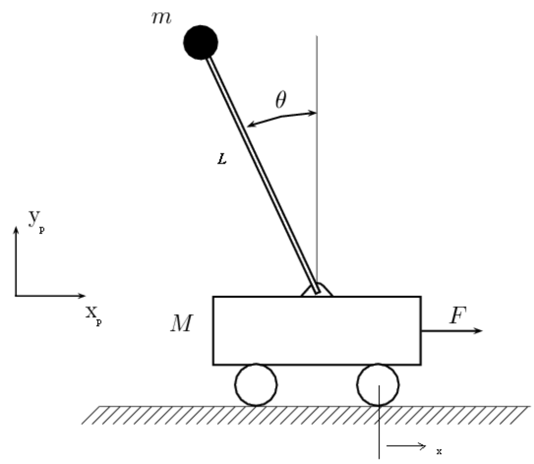

Figure 1 – Basic Inverted Pendulum Diagram

II. PHYSICAL SETUP

The cart is approximately 12 inches by 12 inches with a height of about 8 inches (Figure 2). It has a weight of about 4.8 pounds and a pendulum length of 36 inches. The pendulum is a 1/8-inch diameter stainless steel rod. The cart was constructed with four 4” rubber wheels, two of which are connected to a free rotating axle and two of which were each connected to Hansen 116-1121627D DC motors with 500 CPR encoders. We chose one motor for each wheel to increase the available torque. The Hansen DC motors were chosen to maximum starting torque with a battery in the 12-volt range. Our battery for the motors is a Tattu R-Line 14.8V 1300mAh LiPo battery.

This battery is low cost and lightweight (five ounces), yet delivers a relatively high burst discharge rate of 150 C.

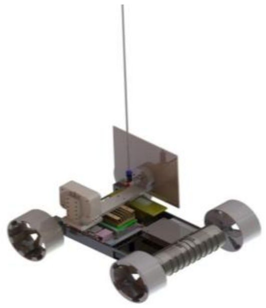

Figure 2: Cart/Inverted Pendulum Design

The virtual sensor was implemented with a Raspberry Pi V2 camera module running at 90 fps in conjunction with OpenCV on an NVIDIA Jetson Nano. The chassis, mounts, and connectors were all designed using Solidworks and created using additive manufacturing techniques. A 5mm diameter red circle was fixed to the pivot point of the pendulum. A blue sphere of the same diameter was placed at an arbitrary location on the rod within the camera frame. Both pieces were manufactured with 3D printed PLA and were sanded to reduce glare. To create a consistent lighting environment, a fixture of seven white LEDs was mounted behind the camera, lighting the area surrounding the pendulum's pivot point.

To power the Jetson, a 10,000 mAh battery was located on the cart. The Pololu High-Power Motor Driver 18v22 was used because of its high current rating and ability to be mounted to an Arduino (used in early prototypes). The motor driver must be able to handle two motors simultaneously with PWM control and withstand loads of 3.7A at 18V.

The motor encoders were read by hardware interrupts on an Arduino Nano. The encoder angle was then communicated to the Jetson serially upon request and converted to distance. Interference from background colors in different environments was found to create significant noise, regardless of the Hue Saturation Value (HSV) filters used. To circumvent this issue, a white backdrop was 3D printed and fixed behind the pendulum's base, isolating the colors of the pendulum pivot from those of the background environment. A white backdrop was found to be the most effective, as other solid colors created noise due to the reflection of red and blue hues. A photo of the setup is provided in Figure 3.

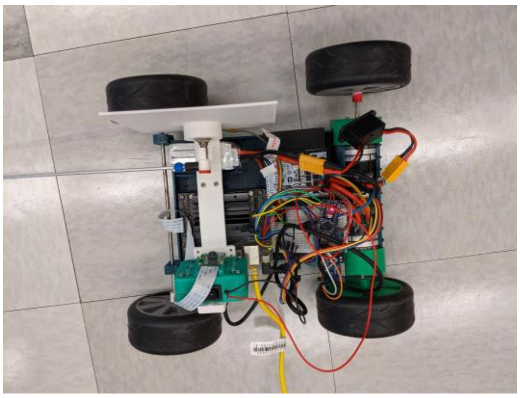

Figure 3 – Photo of Cart and Pendulum

III. VIRTUAL SENSOR DESIGN

The virtual sensor works by locating the center of the red and blue circles (Figure 4) described in Section II and calculating the angle between them. Because the red circle is fixed to the pendulum's pivot point and the blue circle lies at an arbitrary point on the pendulum rod, the angle between these circles taken with respect to a vertical axis is equal to that of the pendulum itself. OpenCV is used to filter the red and blue circles from the background (Figures 5 and 6), find the center of each circle, and then calculate the center coordinates of each shape.

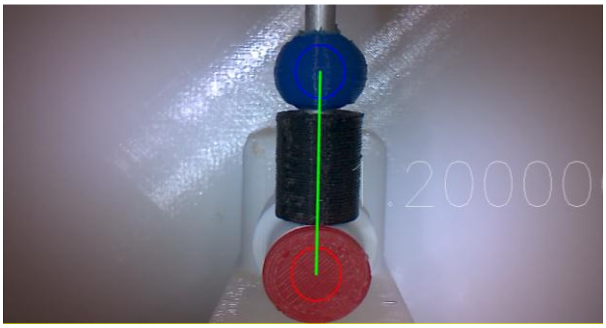

Figure 4 – Picture of rod angle measurement. Angle displayed on the screen in real time.

Figure 5 – Virtual Sensor Filtered Blue Geometry from image in Figure 4.

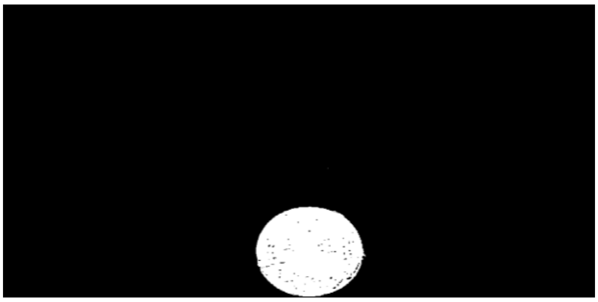

Figure 6 – Virtual Sensor Filtered Red Geometry from image in Figure 4.

The angle can then be calculated with a basic inverse tangent function:

$$\theta = arctan\left(\frac{x_{BLUE} - x_{RED}}{y_{BLUE} - y_{RED}}\right)$$

The white background shield and LED box narrow the bands of HSV seen within the camera frame, making color filtering trivial. The other concern is the loop rate of the sensor: because photo frames are stored in an array, filtering and similar operations can come at a notable time cost. In a control context, even the reduction of the loop rate by a few milliseconds may be significant. As such, the grabbing of individual frames was placed on another thread and, immediately upon reading, camera frames are cropped to the minimal possible size that includes both the red and blue shapes.

Because time delay is critical to the control of the inverted pendulum, we decided to measure both the delay and precision of the virtual signal by comparing it to a standard encoder. We chose a Bourns EMS22A50 rotary encoder which has 1028 counts per revolution. Figure 7 shows both the encoder and the virtually acquired angle measurement. To obtain the signals we simply moved the pendulum in the upright position back and forth.

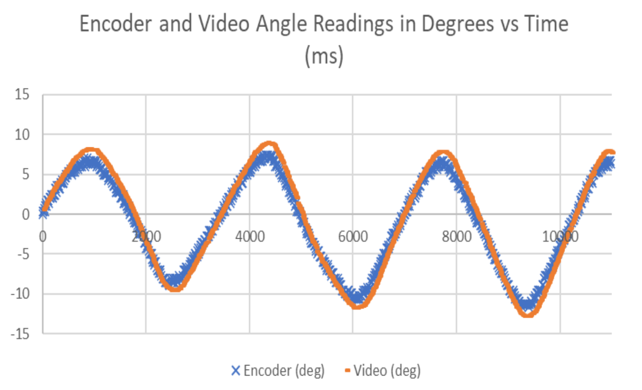

Figure 7 – Encoder and Video Angle Measurements vs. Time in milliseconds.

In isolation, it was found that the visual servoing algorithm took 5.31ms on the Jetson Nano. It takes 3.68ms to filter the cropped 300x160 image at 480x852 resolution. With the simple LQR and motor control algorithms included, the pendulum has an average loop rate of 12ms. While significant, this is typically low enough to remain controllable. The second major concern when utilizing visual servoing is the time lag between the virtual sensor and the physical system. We chose to investigate how far our virtual sensor lags behind a standard absolute rotary encoder, assuming the time delay between the physical system and the encoder is negligible. Using the time difference between the zero-crossings of the two sensors when subjected to a sinusoidal input, such as in Figure 7, the virtual sensor's time delay was calculated to have an average of 4.16ms.

The virtual sensor also maintains a high degree of accuracy for small angles, though the measured angle becomes exaggerated as the magnitude of the true angle increases. This is likely due to the camera's inherent distortion and the error was found to conform to a straight line function. Because our goal is to develop a control system to keep this angle at a minimum, adding this correction factor was decided to be unnecessary for our application.

IV. SYSTEM MODELLING AND CONTROLLER DESIGN

Creating a system model of an inverted pendulum has been well established [6]. In our case, we do not have an end-mass on the pendulum, so we modeled the pendulum as a 36 inch rod with a mass of 22 grams. The cart force is generated by torque from the two motors specified earlier. The PWM voltage signal sent to the motors was found to be linearly proportional with the force on the cart. This was accounted for by multiplying the input matrix of the state space model by the PWM to force conversion factor. It was also assumed that the viscous friction coefficient for the cart was zero, as cart acceleration varied negligibly with cart velocity. Figure 8 shows how the acceleration of the cart is linearly proportional to the input percentage of the motor. The full 100% input is the rated 12V input to the motor.

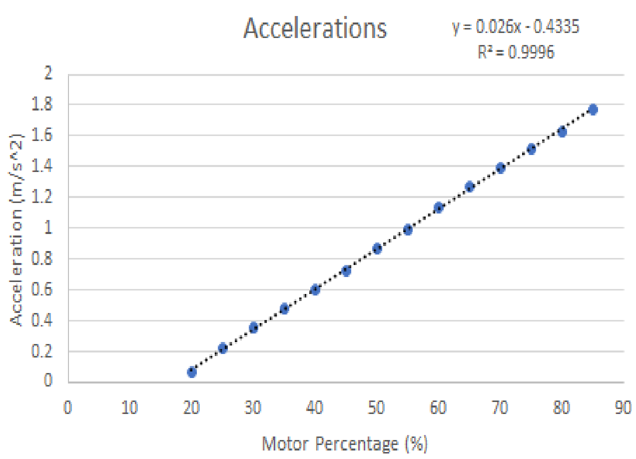

Figure 8 – Acceleration of the cart vs the percentage of the maximum voltage applied to the motor.

By demonstrating that the motor voltage is proportional to the acceleration of the cart, we show that we can linearly control the effective input force on the cart. The observed accelerations of the cart can be used to precisely determine the prescribed input force. Thus, a standard linearized state-space model is valid with this physical setup. Small bends and deformations in the rod that accumulate over time during testing were found to have a negative impact on the performance of the pendulum controller. Before each test, the pendulum rod was balanced and the angle reading was saved by the controller as the rod angle set point, which was then subtracted from subsequent angle readings.

Controller design for an inverted pendulum is also well established in literature [7]. The goal of this paper is to outline how it is now possible to use a video signal as a virtual sensor, not to outline the controller design. A standard LQR controller was developed for the IP. The LQR controller multiplies each of four state variables by a constant respective gain. The four state variables include cart position, cart velocity, pendulum angle, and pendulum angular velocity. A simple MATLAB script (using MATLAB's built-in LQR function) was used to optimize the LQR gain matrix k using the following cost function:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R = 1$$

The controller was implemented in software on the Nvidia Jetson. The cart position was calculated using the angle measured by the motor encoders and the pendulum velocity was obtained by taking the numerical derivative of the virtual angle measurement. Figure 9 shows the pendulum angle as a function of time under LQR control in response to the rod falling due to gravity. The angle motion was greater than expected, but the pendulum continued to be brought back to upright position.

It should be noted that over time the cart position would drift and the LQR gains should likely be adjusted to keep the cart position closer to the starting point. This drift in position can be seen in Figure 10, which shows the angle and position response to an initial angular offset and velocity in the negative direction. Although the cart position does drift, the pendulum can be balanced by the control system, using the virtual angle measurement from the video signal. Figure 11 shows a still photo of the pendulum balancing using the LQR controller. For this trial there was no usage of the angle encoder, the only measurement of the pendulum was from the video signal.

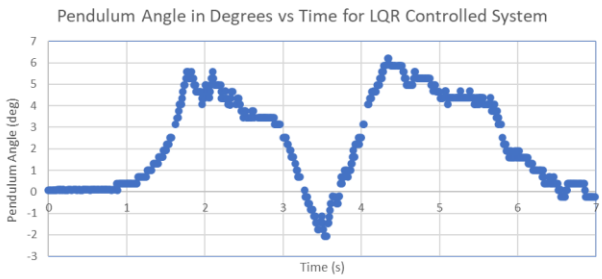

Figure 9 – Pendulum angle as a function of time under control signal using virtual sensor.

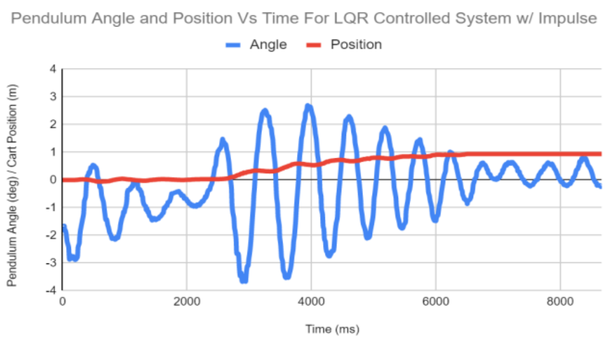

Figure 10: Pendulum angle under control signal using virtual sensor with nonzero initial conditions

Figure 11: Still photo of control system in action. Arrow used to locate pendulum.

V. CONCLUSION

The inverted pendulum is a challenging control problem that requires a very low time delay. Using an Nvidia Jetson and an Arduino, we have demonstrated that it is possible to stabilize the inverted via visual servoing. The processing speed of even relatively inexpensive off-the-shelf components have reached a point where virtual sensors can be effective in the domain of control systems, where time resolution of the sensor is of utmost importance. Similar virtual sensors could be especially effective in areas where the resolution of the measured variable is of more concern than the additional time delay. As component quality and processing speed continue to improve, virtual sensors may be a reasonable solution for a greater variety of applications. It is expected that video processing technology will continue to enable faster controllers and will likely reshape many control applications in the future.

REFERENCES

[1] D. Wu, S. Ci, H. Luo, Y. Ye and H. Wang, "Video Surveillance Over Wireless Sensor and Actuator Networks Using Active Cameras," in *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2467-2472, Oct. 2011, doi: 10.1109/TAC.2011.2164034.

[2] R. C. Seals, "Position and orientation sensor using multiple video cameras," IEE Colloquium on Advances in Sensors, 1995, pp. 7/1-7/5, doi: 10.1049/ic:19951511.

[3] N. Abbas and F. Yu, "A Traffic Congestion Control Algorithm for Wireless Multimedia Sensor Networks," 2018 IEEE SENSORS, 2018, pp. 1-4, doi: 10.1109/ICSENS.2018.8589923.

[4] A. Xhafa, P. Tuset-Peiro and X. Vilaiosana, "Live demonstration: Wireless PID control of a thermal process using an ultra-low cost LWIR camera," 2017 IEEE SENSORS, 2017, pp. 1-1, doi: 10.1109/ICSENS.2017.8234030.

[5] M. Landry, S. Campbell , K. Morris, and C. Aguilar, “Dynamics of an Inverted Pendulum with Delayed Feedback Control,” SIAM Journal of Applied Dynamical Systems, 2005, Vol. 4, No. 2, pp. 333–351.

[6] “Controls Tutorial for Matlab and Simulink”, University of Michigan, Carnegie-Mellon, and Detroit Mercy, https://ctms.engin.umich.edu/CTMS/index.php?aux=Home.

[7] Prasad, L.B., Tyagi, B. & Gupta, H.O. Optimal Control of Nonlinear Inverted Pendulum System Using PID Controller and LQR: Performance Analysis Without and With Disturbance Input. Int. J. Autom. Comput. 11, 661–670 (2014). https://doi.org/10.1007/s11633-014-0818-1