Vision Based Balancing Tasks for the iCub Platform: A Case Study for Learning External Dynamics

Stephen E. Levinson, Aaron F. Silver, Luke A. Wendt
Department of Electrical and Computer Engineering – The University of Illinois at Urbana-Champaign

***Abstract* - Dr. Stephen Levinson's Language Acquisition and Robotics research group is currently engaged in research in the development of intelligent robots and their ability to learn natural language. Language is acquired through interaction with the real world, where sensory-motor function is essential. To answer questions on sensory-motor function, a variety of balancing tasks are being considered. These tasks incorporate an unstable external dynamic with a clear measure of success and failure.**

I. Introduction

We wish to achieve control of an unstable external dynamic through vision feedback on the humanoid iCub platform via machine learning. Before the problem is approached with machine learning, a feasibility assessment needs to be done to verify the task can be accomplished. Any hardware or software limitations need to be identified. This exercise serves to familiarize us with the hardware as well.

The overarching goal of this research is to combine learned motor sensory control in an associative memory framework. From there, the robot can begin to learn via spatial reasoning. This is a process we believe to be fundamental behind language acquisition.

In order to reduce complexity, a balancing task was sought with a slow dynamic that did not require joint torque feedback. For these reasons, we chose to begin with balancing a ball on a plate. The objective of the ball & plate system is to maintain a rolling ball in the center of a flat plate through dynamic reorientation of the plate.

To implement this system, a plate was constructed with a perpendicular handle underneath for the iCub to grasp. The iCub was then able to visually identify the position of the ball in relation to the plate and interact with the ball through orientation of its wrist.

II. Ball Dynamics

A. Ball & Beam

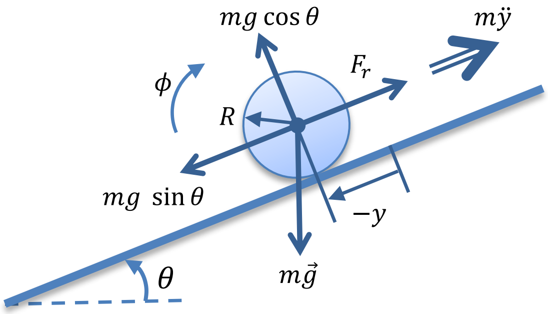

Fig. 01. Free Body Diagram of Ball & Beam System

Table 1
Model Parameters

Variable	Description	Units
$\boldsymbol{m}$	Mass of the Ball	kg
$\boldsymbol{R}$	Radius of the Ball	m
$\boldsymbol{g}$	Gravitational Constant (≅ 9.81)	m/s^2
$\boldsymbol{J}$	Rotational Inertia of Ball	$kg\ m^2$
$\boldsymbol{y}$	Distance of Ball from Center of Beam	m
$\boldsymbol{\phi}$	Rotation Angle of the Ball	rad
$\boldsymbol{\theta}$	Roll angle of Wrist	rad
$\boldsymbol{y_r}$	Reference Distance from Center of Beam	m
$\boldsymbol{\theta_{eq}}$	Angle where Beam is Perpendicular to $\vec{g}$	rad
$\boldsymbol{b}$	Control Gain $= g/(1+\mu_j)$	N/rad
$\boldsymbol{w}$	Constant Disturbance Input to Ball	rad
$\boldsymbol{b_f}$	Viscous Friction Coefficient	$N/(m/s)$
$\boldsymbol{c_f}$	Coulomb Friction Coefficient	N
$\boldsymbol{\mu_1}$	Inertia Coefficient for Solid Sphere (2/5)	-
$\boldsymbol{\mu_2}$	Inertia Coefficient for Hollow Sphere (2/3)	-

Before tackling the ball & plate system, a ball & beam system was considered with the same dynamic and a simpler vision component. The ball & beam system consists of a smooth rigid ball in normal contact with a smooth rigid beam. The free body diagram of the ball & beam system can be seen in Fig. 1. Control is introduced by tilting the beam. The model parameters are described in more detail in Table 1.

The projection of gravitational force along the beam accelerates the ball. The projection force is related to the angle of the beam. This becomes the control input to the system. Any acceleration of this angle produces normal forces on the ball with the beam, which do not result in acceleration along the beam. This property allows a stabilizing balance to be implemented on the system without requiring torque feedback on the beam. Assuming the ball is of negligible mass, its influence on the inertia of the beam can be ignored.

Define the ball inertia $J_j = \mu_j m$ where $\mu_1 = 2/5$ for a solid sphere and $\mu_2 = 2/3$ for a hollow sphere. The hollow sphere provides a slower dynamic for study. The resulting ball dynamic is

$$\ddot{y} = \frac{g}{1+\mu_j}\sin\theta \approx \frac{g}{1+\mu_j}\theta. \qquad (1)$$

This assumes knowledge of the operating point θ_{eq} where $\ddot{y} = 0$. Assume this exact angle is not known. This can be modeled by adding a constant disturbance w. The last piece to add to this model is friction. Adding viscous and coulomb, friction the ball dynamic becomes

$$\ddot{y} = b(u+w) + \underbrace{b_f\dot{y} + c_f\ \mathrm{sign}\ \dot{y}}_{friction}, \qquad (2)$$

as seen in Fig. 02.

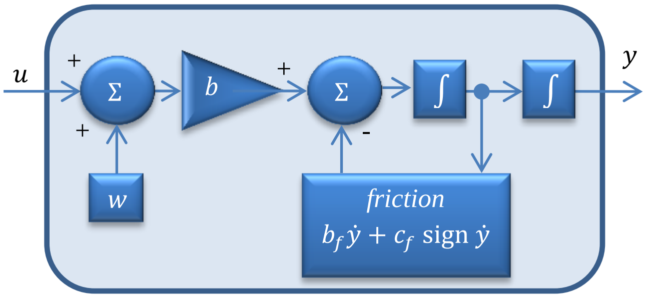

Fig. 02. Ball Dynamic

B. Ball & Plate

The ball & plate dynamic is a simple extension from the ball & beam system. The ball & plate dynamic acts independently on the ball in two directions:

$$\begin{aligned} \ddot{y}_1 &\approx b(u_1 + w_1) + b_f \dot{y}_1 + c_f \text{ sign } \dot{y}_1 \\ \ddot{y}_2 &\approx b(u_2 + w_2) + b_f \dot{y}_2 + c_f \text{ sign } \dot{y}_2. \end{aligned} \tag{3}$$

Fig. 03 shows the ball & plate setup. Here $u_1 = \theta_1$ is the wrist roll, which corresponds to the same angle used in the ball & beam system. The other angle is $u_2 = \theta_2$ and corresponds to the pitch of the wrist.

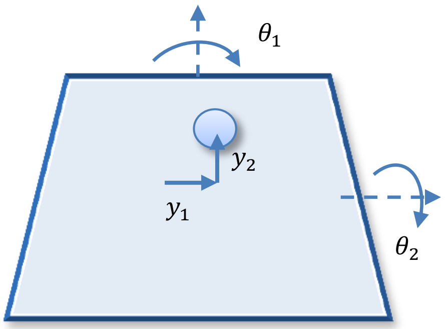

Fig. 03. Plate & Ball

III. System Overview

There are three major components left to address. The first component involves determining the motor dynamic. This address how faithfully the beam or plate can reach the desired configuration θ_r. For the sake of simplicity, the inertia of the hand and beam or plate will be considered part of the motor inertia.

The second component composes the capabilities of the vision system. This includes resolution, frame rate, noise, distortion, and how they can be overcome to faithfully estimate the position of the ball in relation to the plate.

The last component involves designing a control that takes all these dynamics into account, and then finds a way to track a reference position r as faithfully as possible despite all the limitations of the system and remaining uncertainties. This system layout is summarized in Fig. 04.

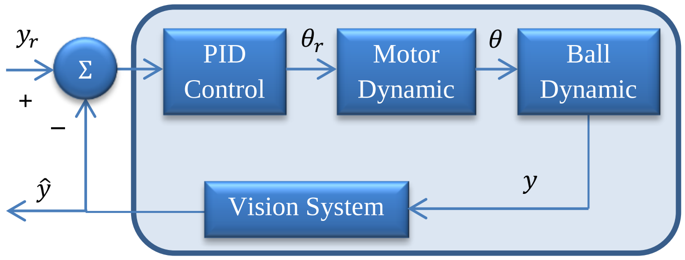

Fig. 04. System Overview

IV. Motor Dynamic

A. Structure

A few things were known about the motor beforehand. A small amount of elasticity is observed in the hand in ways that cannot be measured by the encoders on the hand. Each motor is controlled by an onboard DSP operating at a higher frequency than the computer terminal. The DSP uses PID control that is influenced by five parameters. These parameters can be updated no faster than the sampling time of T_s on the computer terminal. The first three parameters are the PID gains. These gains were left unaltered in all experiments. The last two parameters are the desired reference position of r for the encoder and the desired velocity of V for the motor. That is the system will arrive at the desired Δr position in $\Delta r / V$ seconds. To achieve this task, a 5th order polynomial reference is generated with zero velocity and acceleration at the terminal points. There is then some unknown closed loop feedback dynamic in response to these parameters as summarized in Fig. 05.

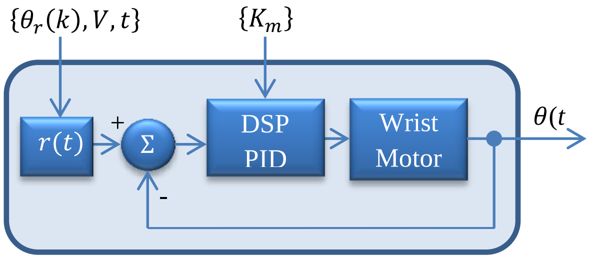

Fig. 05. Unknown Motor Dynamic

B. Identification

A large sample of target ranges and speeds with the wrist roll motor were recorded for study. The motor was observed to faithfully track a 5th order polynomial at speeds up to 30 deg/s with a range of -20 deg to 20 deg. This will be the complete motor dynamic for all modeling. The motor Dynamic block is represented in Fig. 06. The motor is essentially a rate limiter where $\Delta_r \leq V T_s$.

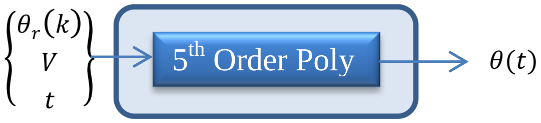

Fig. 06. Motor Dynamic Approximation

V. Vision System

A. Beam System

Direct ball position is not available to the input of the controller in this system. Instead, the cameras in the head of the robot were used to visually approximate the position of the ball. To detect the length of the beam, two different colored balls (pink and orange) were placed on the end of the beam. A green ball would then be placed on the beam that would be free to run along the length of the beam as seen in Fig. 07.

The robot's right camera was centered on to the ball & beam system. The image plane coordinates in pixels were not transformed to correct for projection error. The projection error was assumed to be minimal in this circumstance. The camera ran with a resolution of 320x240 pixels/frame and a frame rate of approximately 20 frames/second.

For each frame, color thresholding was done for object detection, since each ball had a unique RGB value that was not found in the background. Each pixel was assigned to either the green, pink, or orange ball on an initial raster scan. A second raster scan then proceeded, and if a pixel was flagged as a certain color, a 7 pixel neighborhood around it was sampled. If a sufficient number of the pixels in the neighborhood were also flagged as the same color (a threshold value set to 20 pixels), this pixel is assigned to be part the object of that color ball. This extra neighborhood step was done to produce an accurate measurement and reduce noise.

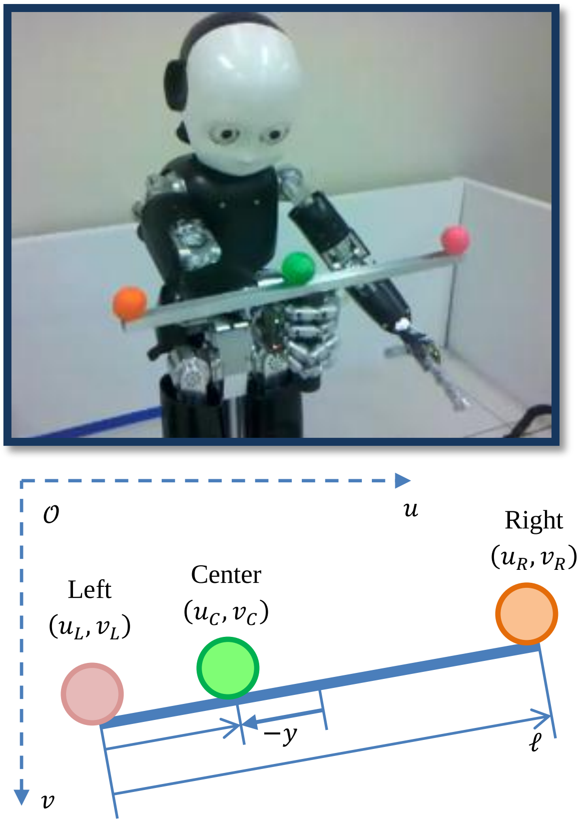

Fig. 07. Ball & Beam Vision System

The centroid of each object was computed. Once each centroid was found, the Euclidean distance between the green ball and each of the two colored balls on either end was calculated. The difference of these two calculations then represented the offset of *y* the green ball had from the center of the beam. Using the fact that the beam was exactly 33.5cm from one end to the other, it was possible to convert pixels into actual length measurements of centimeters.

To determine the image noise, $\hat{y}$ was recorded in a stationary state for 60 seconds. The error appeared Gaussian. The variance was found to be $4.7 \cdot 10^{-4}$ cm^2. The image error can be expected to be under 0.1cm. The average sample time was 0.06 sec with a variance of $5 \cdot 10^{-4}$sec^2. This includes the in loop reference command to the motor and return encoder values. Fig. 08 summarizes these results.

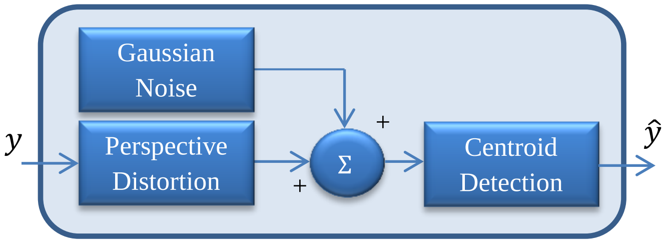

Fig. 08. Model of Vision System

B. Plate System

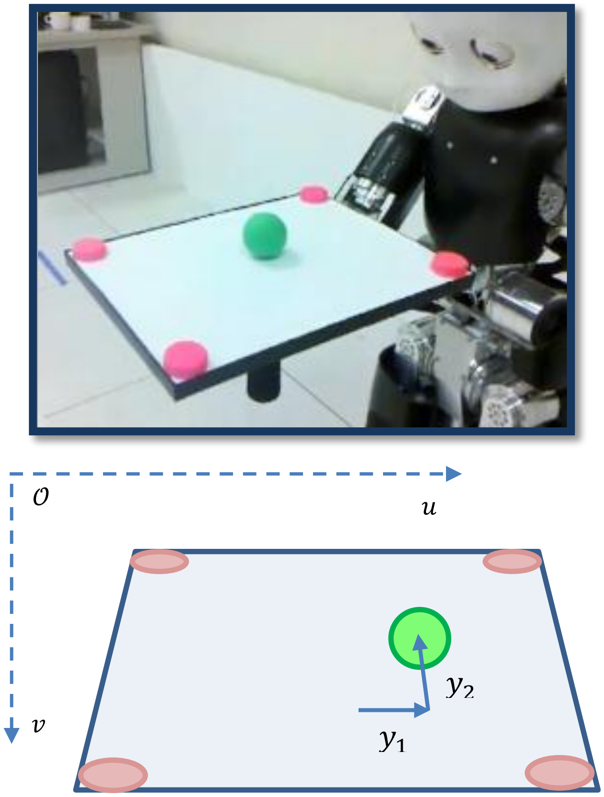

Fig. 09. Ball & Plate Vision System

The ball & plate system can be seen in Fig. 09. When wishing to expand into another dimension, more sophisticated computer vision procedures were required to accurately gather ball position. Once again, only the right camera was used. The robot camera sees a skewed image of a flat rectangular surface. In order to take the skewed image of the plate and turn it into a top down image of the plate, a projection mapping was applied to the image plane.

A homography projection transformation was selected to make this correction. Homography is a mapping between points of one plane to another under projective transformation through a single point. It creates a 3x3 transformation matrix. To perform the transformation four points were required. These points were taken from the corners of the plate using the same centroid finding procedure mentioned earlier.

VI. Control

A. *State Feedback with Integral Action*

To implement control, state feedback with integral action was utilized. This type of control was chosen due to its robust disturbance rejection and ability to track quadratic functions. To minimize the effects of high frequency noise from the vision system, a first order low pass filter was added to the derivative calculation. A Linear Quadratic Regulator (LQR) was used as a heuristic design tool for pole placement. The primary goal was simple asymptotic stability with settling time of around 10 sec and overshoot below 50% to step inputs. This control structure is summarized in Fig. 10.

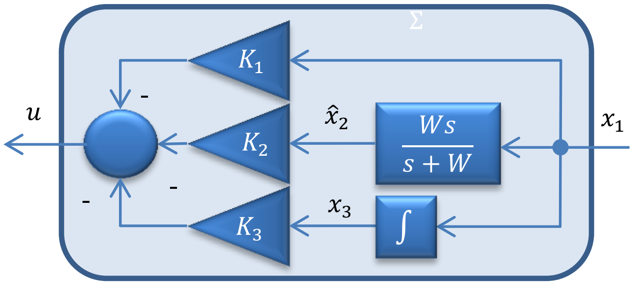

Fig. 10. Filtered State Feedback with Integral Control

B. *Discrete Implementation*

Using Tustin's Method, the continuous filtered control system was converted into a discrete control system operating at a sampling rate of T_s. The Tustin mapping is

$$s \to \frac{2}{T_s}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right), \tag{4}$$

were z^{-1} represents a time delay of T_s and the continuous system is converted into a difference system. The Simulated performance of the continuous and discrete control can be seen in Fig. 11 where Δ denotes the discrete implementation.

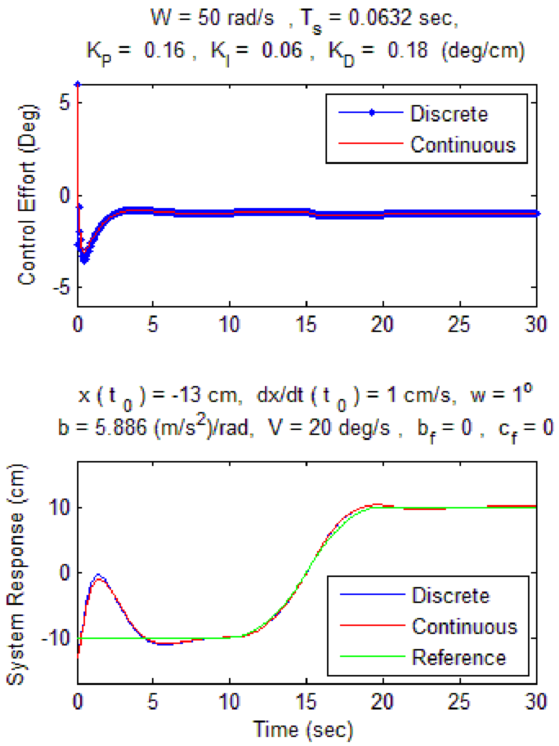

Fig. 11. Simulated Quadratic Reference Tracking

VII. Results

A. *Implementation of Ball & Beam on iCub*

The remaining pieces of the model were added to the system model given by Fig. 04. The observed Motor Dynamic given in Fig. 06 was added so that on every sample iteration a 5th order reference polynomial of velocity V was sent to the target position. A velocity of $V = 30$ deg/s was used in both simulation and hardware. The vision system model given in Fig. 08 was added. Just the Gaussian noise on the observer was added with no perspective distortion. The control input was saturated at $\pm 6°$ and the ball position on the beam was saturated at ± 17cm.

The refined model was observed to perform well to the same control found in section VI with $W = 50$ rad/s and $K^* \cong (0.16 \quad 0.18 \quad 0.06)^T$ deg/s. The Bode plot produced by the control loop acting on the ball had a phase margin of 60 deg at a cross over frequency of 1.94 rad/s. The sampling rate used by the system was sufficiently slower than the maximum delay allowable without losing stability.

These control parameters were then used on the ball & beam system, and the iCub successfully stabilized a ping pong ball. The comparison of the model to the experimental data is given in Fig. 12. The ping pong ball responded a little slower with less overshoot.

There is still a small discrepancy between the model and the data, but the control clearly worked as desired. The source of most of the discrepancy is most likely the friction acting on the ping pong ball. The ball also has a seam on it that makes its density unevenly distributed causing rocking motion.

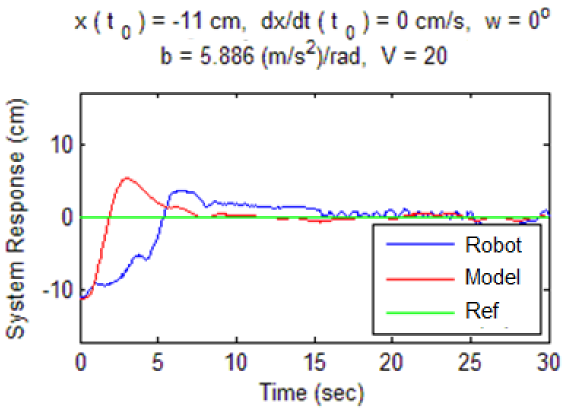

Fig. 12. Comparison of Simulation to Experimental Ball & Beam Data

B. *Implementation of Ball & Plate on iCub*

Using the same control found on the ball & beam system, the plate successfully stabilized the ping pong ball from the front left corner of the plate to the center of the plate as seen in Fig. 13. The source of the disturbance at 20 sec was most likely from the inhomogeneous makeup of the ball. The settling time and overshoot are very satisfactory.

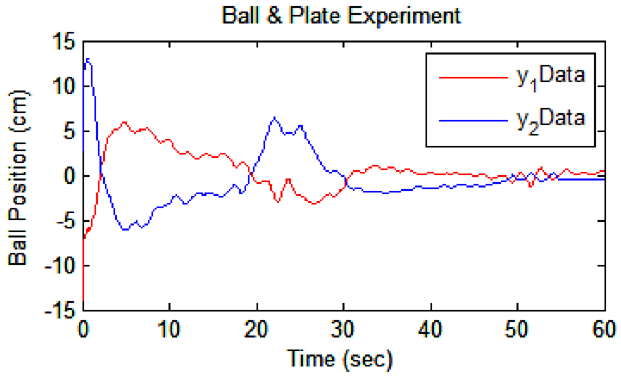

Fig. 13. Experimental Data from Ball & Plate Stabilization

VIII. Conclusions and Future Work

We are very happy with the results achieved so far. At present, we have implemented no machine learning with this problem. That will be included in future work. Now that feasibility has been demonstrated, it will definitely be an exciting challenge to teach the robot to perform the same task on his own. The primary benefit of this project has been to gain familiarity with the software and hardware on the iCub.

In the near future, we will incorporate the kinematics of the robot so that groups of motors may be utilized in more complex tasks such as balancing an inverted pendulum in the iCub's hand. The ultimate goal of this research is to incorporate different types of learned motor control into an associative memory framework hopefully resulting in new and interesting intelligent behavior.

Acknowledgment

We would like to thank the other members of the UIUC Language Acquisition and Robotics research group: Alex Duda, Lydia Majure, and Logan Niehaus. Also we would like to thank Dr. Giorgio Metta as well as the Italian Institute of Technology for providing us with the iCub Robot.

References

[1] Franklin, Gene. Feedback Control of Dynamic Systems, 6th Edition 2009

[2] Bishop, Robert H. and Richard C. Dorf. Modern Control Systems. Prentice-Hall, 2005.

[3] Hartley, Richard and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.

[4] "Manual - Wiki for RobotCub and Friends" http://eris.liralab.it

[5] Forsyth, David and Derek Hoiem. UIUC-ECE 549 course notes

[6] Basar, Tamer, Sean Meyn and William Perkins. UIUC-ECE 515 course notes