Using Floating-Gate Based Programmable Analog Arrays for Real-Time Control of a Game-Playing Robot

Scott Koziol, David Lenz, Sebastian Hilsenbeck, Smriti Chopra, Paul Hasler, and Ayanna Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
skoziol, phasler, ayanna.howard@ece.gatech.edu, LenzDavid@gmx.de, sebastian.hilsenbeck@gmail.com, smritichopra3@gmail.com

***Abstract*—This paper presents preliminary results of a mobile manipulator robot tasked to play the classic Towers of Hanoi game. We first discuss the control algorithms necessary to enable necessary game-playing behavior and provide results of implementing our methodology in a high fidelity 3D environment. After attaining success in the simulation environment, we provide results on implementation of the same control software using physical robot hardware. Additionally, preliminary analysis for implementing analog Proportional-Integral-Derivative (PID) control on this platform using a floating-gate based reconfigurable analog IC is explored. Using this concept of floating gate analog arrays for control enables off-loading of the processing, which could be helpful for real-time implementation of robot behavior.**

***Index Terms*—Robotics, Towers of Hanoi, Pioneer, Player, Gazebo, FPAA, PID, floating-gate, OTA**

I. The Mobile Manipulator and FPAA

A mobile manipulation robotic platform solving the Towers of Hanoi puzzle is described in this paper. This preliminary work is described with the possibility of leveraging this work in two domains: First, the mobile manipulator could be developed into a system that could interact with children or adults in a *turn taking* scenario. Second, this platform can be used to investigate feedback control systems implemented with reconfigurable analog electronics. Robot control software called *Player* is used as the main software for this system, [1], [2]. *Player*, running on a laptop, is the brains of the system. It receives sensor input from an overhead camera for localization and then commands the robot as desired. We operate *Player* in two robot environment modes. The first mode is *Player* interacting with a real robot in the real world. The second mode is *Player* interacting with a simulated robot in a 3D simulated environment with dynamics. This 3D environment is called *Gazebo*.

The *Player* software has the ability to interact with a reconfigurable analog electronics system called a Field Programmable Analog Array (FPAA). Fig 7 shows the FPAA and programming and control hardware infrastructure to be used [3]. This embedded system is controlled from Matlab and uses a USB connection to communicate with a microcontroller on the circuit board. In this configuration, the FPAA could be characterized as a *Feedback Control Co-processor* for the robot's navigation system. Path planning is another demonstrated use of an FPAA in robotics [4].

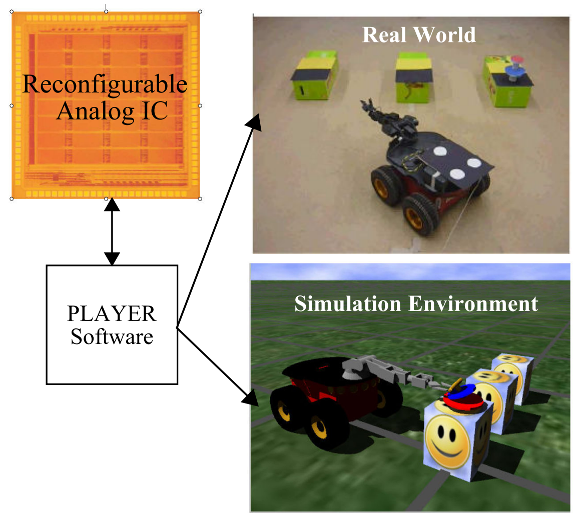

Fig. 1. This is the big picture of the system: A client software called *Player* interacts with either the real world or a simulated world and solves the classic Towers of Hanoi puzzle. Additionally, the software has the ability to interact with a reconfigurable analog co-processor.

The mobile manipulator system, using a Pioneer robot and arm [5], is demonstrated solving the classic Tower of Hanoi problem, Fig 1. In this puzzle, a tower of disks is created by stacking disks on top of each other. One of the rules is that only smaller disks may be placed on larger disks. This version assumes there are three possible locations for the tower's location. The tower starts in one of these locations. The goal is to move the tower from one location to another location.

This robotic manipulator has three main tasks: Sensing, Thinking, and Acting. The *Sensing* task involves an overhead camera as the primary sensor. Image processing tasks for the Towers of Hanoi problem include segmenting the disks from the background and identifying their size and position. The

Thinking tasks include creating a sequence of legal actions for moving the disks so that the goal is achieved (path planning), as well as turning these high level commands into low level control functions. The *Acting* tasks includes commanding the Pioneer robot's forward/reverse velocity and rotation as well as commanding an attached five degree of freedom (DOF) Pioneer manipulator arm to move the pieces.

Section II discusses related work, Section III describes architecture for Sensing, Thinking, and Acting, Section IV compares differences between simulation and real world operation, and Section V is a closing summary.

II. Related Work

A. Playing with Robots

Robots have been used in the past for games such as chess [6], or as a therapy aid [7]. Robots have also been used to help children with disabilities [8], [9].This mobile manipulator could be extended for use in the future work such as *turn taking*, [10]. In [11] a simple non-mobile manipulator is described for solving the Towers of Hanoi problem. This was part of a Robotics Education Lab at CMU. In [12] humans used a mobile web interface to instruct a PR2 how to solve the towers of Hanoi problem. A video of a PR2 and many other robots solving the Towers of Hanoi may be found on the internet.

B. Analog Control

A number of recent papers have been written regarding using reconfigurable analog circuits called Field Programmable Analog Arrays (FPAA) for low level control.This paper and [13] are based around custom FPAAs, but many are based on the switch-capacitor Anadigm IC design [14], [15], [16]. General references concerning PID controllers are [17], [18], [19], and [20]. Background for using Operational Transconductance Amplifiers (OTAs) for PID control is found in [21] and [22]. Finally, although this robotic system is accessible and easy upgraded and serviced, this is not always the case for all robotic platforms. Other FPAAs are being explored to allow flexibility in sensing and control circuits of space systems [23], [24]. The FPAA in this paper is typically different than other reconfigurable analog circuits because it uses floating-gate transistors as the switch matrix.

III. Architecture for Sensing, Thinking, and Acting

One of the goals of the architecture is to give the robot a high level of autonomy. The robot's a priori knowledge consists of the following:

- A list of potential disk colors.
- An initial estimate of pole positions.
- The height of the disks.

The system block diagram in Fig 7(c) provides a high level view of the robot's Navigation system and also shows how it interfaces with the planner, vision sensor, and robot hardware. The Sensing, Thinking, and Acting portions of this block diagram are individually addressed in the remainder of this section.

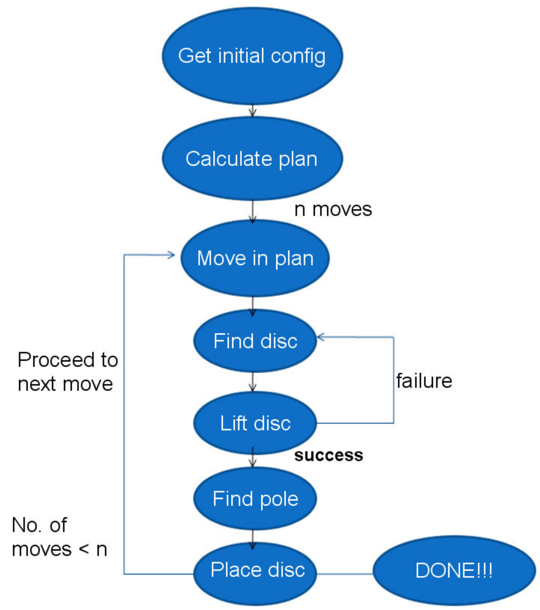

Fig. 2. This figure shows a high level flowchart of the Thinking tasks

A. Sensing

Vision is the primary sensor in this system. It sends information to the Tracker sub-block. It assumes that there is an overhead camera available to image the robot, poles, and disks at all times. "OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision [25]." It has been integrated into the control program for image processing tasks. Fig 8(a) shows an example image from the overhead camera modeled in Gazebo, and Fig 8(b) shows a view from the real overhead camera. Working with the Tracker, this image system is able to successfully segment images using color features and is able to extract colored circles from images.

B. Thinking

The section of the robot's system block diagram that describes *thinking*, Fig 7(c), consists of four main tasks: Navigation, Planning, Tracking, and maintaining the internal World Model. A high level state machine description is found in Fig 2. The first state in Fig 2 is "Get Initial Configuration". In this step, the system determines the number and color of the disks and the initial positions. The a priori information that helps this process is that it is assumed that the disk colors come from a known set of colors in a color list.

The Planner's task is to identify a sequence of actions that will accomplish the goal of moving the disks from their starting position to the goal position. We integrated a previously existing Towers of Hanoi planner into our system, [26]. A plan has the following form:

1) Take the disk on pole 1 and place it on pole 3
2) Take the disk on pole 1 and place it on pole 2
3) Take the disk on pole 3 and place it on pole 2
4) ...

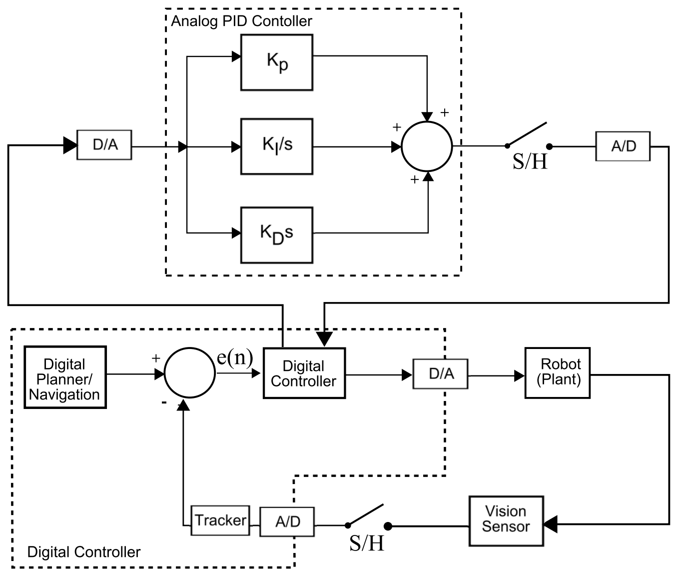

Fig. 3. This figure illustrates an example of how an analog co-processor PID controller could be merged with the digital controller for initial testing. The control system implemented for this project uses a Digital Proportional-Derivative closed loop control system to control the robot's position and orientation.

The Navigation block's task is to convert high level plans to low level commands. Proportional-Derivative closed loop control systems are used to control the robot's angle and forward/reverse position. A block diagram of a Proportional-Integral-Derivative closed loop control system is found in Fig 3. The system was operated using the Digital Controller, but this figure also shows a diagram of how the FPAA based analog PID controller could be integrated into the loop. Ideally, the PID output signal would be sent directly to the plant and not use the A/D and D/A functions.

The Tracker has three main image processing tasks: To determine the Disk poses, Robot pose, and Pole poses. The tracker uses colors to identify objects. To track the disks, first they are segmented from the background with thresholding in the HSV color space. A "blobfinder" is then applied to the segmented image [27]. The blobs are then filtered based on size to determine if they are too large or too small. Finally, the blob's features such as position, area, and standard deviation are calculated and this information is returned to the Navigation routine. This is illustrated in Fig 4. The same process is used to track the poles (boxes) on which the disks sit, except that before the blobfinder is applied the segmented image undergoes erosion and dilation to remove the eyes and mouth of the smiley on the boxes in the simulation. (This process was not used with the real hardware because uniform colored black boxes were used for the poles.) Finally, Robot pose is determined by using a triangle formed by three white dots added to the back of the Pioneer robot. These dots are segmented by the tracker and the robot's pose is calculated. All calculations are in camera coordinates. An internal World Model is also maintained by the robot. This World Model contains three items:

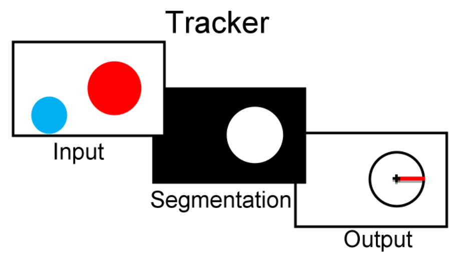

Fig. 4. This figure illustrates that the Tracker first segments the image based on color (In this example it was asked to track the red disk). It then calculates the radius of the disks

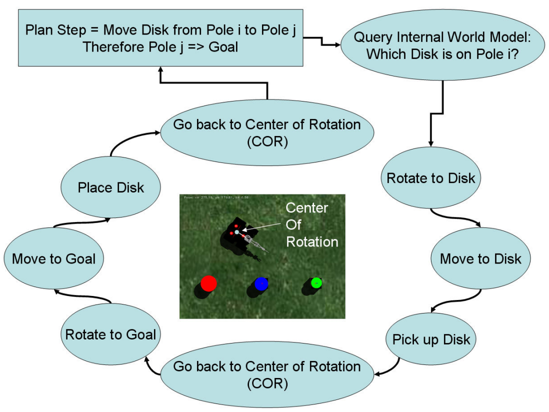

Fig. 6. This figure illustrates the overall guidance and control strategy. The robot will perform this loop for each high level command in the planning sequence.

- List of disks (with each disk's color, position, and radius)
- List of positions of the poles
- Color list

The overall strategy for executing a high level command is shown by the state machine in Fig 6. The robot uses the closed loop controller when rotating to the disk or goal and when moving to the disk or goal.

This software/hardware platform offers a unique capability to integrate our FPAA system into this robot for control. Fig 7(c) shows how the FPAA might be integrated into the system block diagram. The FPAA contains many OTAs. Fig 5(a) shows how OTAs can be used to implement a PID controller [28]. Fig 5(a) builds upon the OTA PID model in [28] by adding parasitic capacitances that are inherent when routing circuits on an FPAA. The current out of an OTA is a function of its transconductance gain, G_m, and the difference between the positive and negative terminals, (2) [29]. Ideally, the current into the positive and negative terminals of an OTA is zero. In subthreshold operation, the output current of an OTA is shown in (1) [29].

$$I_{out} = I_{bias} \tanh\left(\frac{\kappa}{2U_t}(V_p - V_n)\right) \quad (1)$$

For small values, $\tanh(x) \approx x$, and G_m, the so called *transconductnace* of the amplifier, is the slope of the *tanh* curve at the origin.

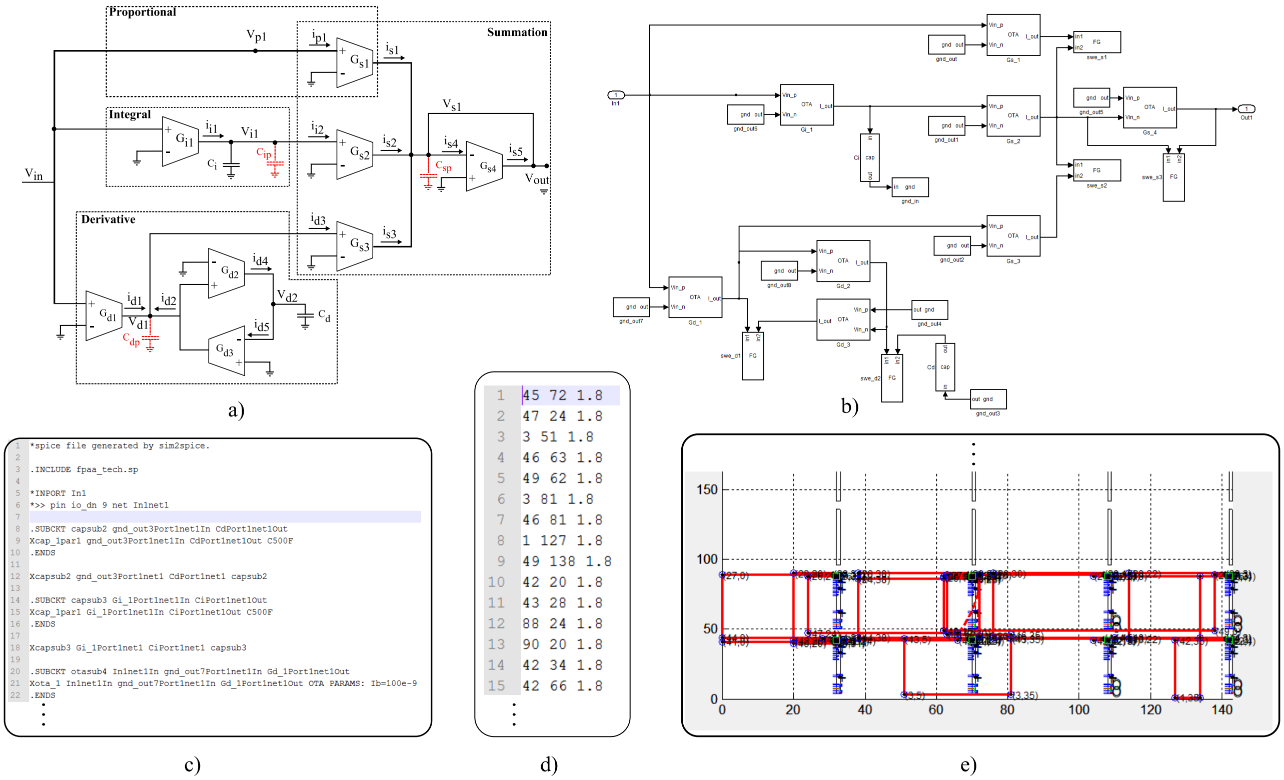

Fig. 5. Design Flow for an OTA based PID controller (a) OTA based PID controller based on [28]. Unlike [28], this model includes parasitic capacitances that are a part of an actual implementation and effect performance. (b) Simulink Block Diagram of controller (c) SPICE list generated by *Sim2Spice* tool (d) FPAA switch list generated by GRASPER tool (e) RAT Figure showing switch list routing on RASP 2.8a IC

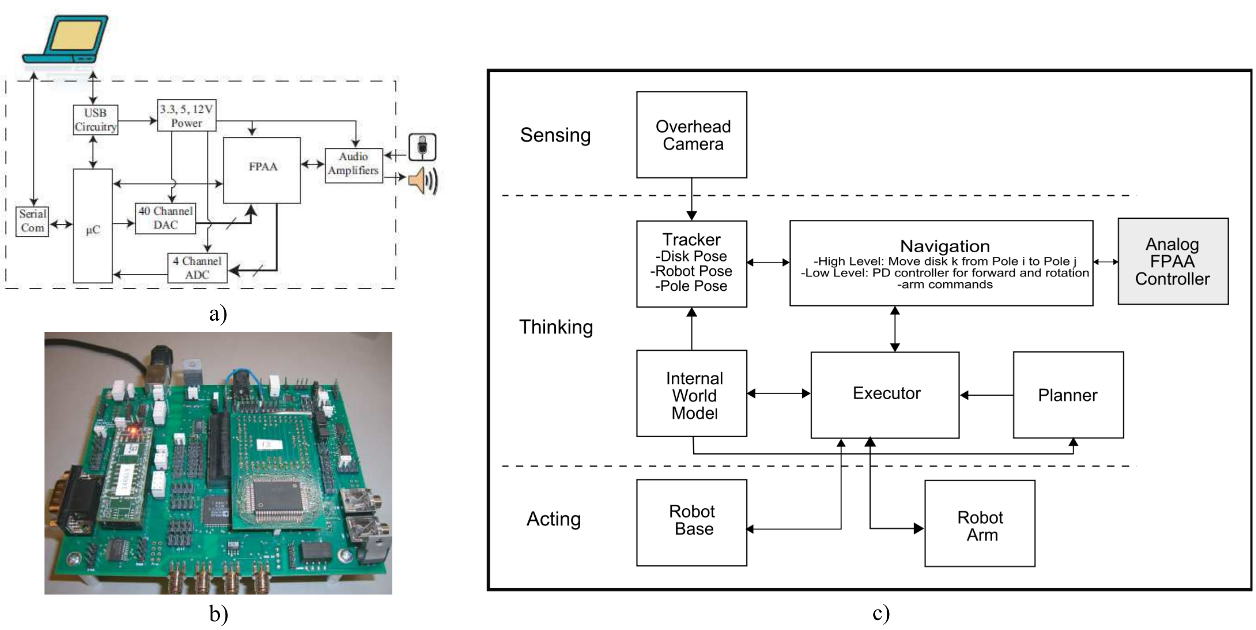

Fig. 7. (a) Block Diagram of the FPAA programming and control board of Fig 7 (b). The board has been designed to be self contained and portable, only needing a laptop. The power and communication is supplied by the USB port. The microcontroller (μC) is a 40 pin DIP plug-in module which uses an ATMEL 32 Bit ARM processor. The FPAA I/O can be reconfigurably connected to the discrete ADC and DACs using headers and jumpers (25.76 square inches) (c). High level control System Block Diagram: This figure shows how the sensing, thinking, and acting systems are combined and where the analog co-processor fits into the larger robot system. [3]

$$I_{out} = G_m\left(V_p - V_n\right) \quad (2)$$

Where G_m is calculated to be:

$$G_m = \frac{\partial I_{out}}{\partial V_{in}} = I_{bias}\frac{\kappa}{2U_t} \quad (3)$$

Therefore, one may adjust an OTA's *transconductnace* by adjusting the bias current, I_{bias}. Using the notation from Fig 5(a), the PID gains K_P, K_I, and K_D in Fig 3 for an OTA based controller are as follows. The intermediate *Proportional* voltage term is:

$$V_{p1}\left(s\right) = V_{in}\left(s\right) \quad (4)$$

The intermediate *Integral* voltage term, taking into account integral circuit parasitic capacitance, C_{ip}, is:

$$V_{i1}\left(s\right) = \frac{G_{i1}}{C_i + C_{ip}}\frac{1}{s}V_{in}\left(s\right) \quad (5)$$

The intermediate *Derivative* voltage term, taking into account derivative circuit parasitic capacitance, C_{dp}, is:

$$V_{d1}\left(s\right) = \frac{G_{d1}}{C_{dp}s + \frac{G_{d2}G_{d3}}{C_d s}}V_{in}\left(s\right) \quad (6)$$

The individual PID currents are added using four OTAs. Taking into account summation circuit parasitic capacitance, C_{sp}, the equation is:

$$G_{s1}V_{p1} + G_{s2}V_{i1} + G_{s3}V_{d1} + I_{out} = C_{sp}\frac{dV_{out}}{dt} \quad (7)$$

$$I_{out} = -G_{s4}V_{out} \quad (8)$$

Substituting (8) in to (7), taking the Laplace transform, and simplifying yields a transfer function for the analog PID controller with parasitic capacitances:

$$\frac{V_{out}\left(s\right)}{V_{in}\left(s\right)} = \frac{1}{\left(C_{sp}s + G_{s4}\right)}\begin{pmatrix} G_{s1} \\ +\frac{G_{s2}G_{i1}}{C_i + C_{ip}}\cdot\frac{1}{s} \\ +\frac{G_{s3}G_{d1}}{C_{dp}s + \frac{G_{d2}G_{d3}}{C_d s}} \end{pmatrix} \quad (9)$$

TABLE I
ANALOG PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER DESIGN WITH AND WITHOUT PARASITIC CAPACITANCES

Gain Term	Ideal	Realistic with parasitic capacitance
Proportional (K_P)	$\frac{G_{s1}}{G_{s4}}$	$\frac{G_{s1}}{\left(C_{sp}s+G_{s4}\right)}$
Integral (K_I)	$\frac{G_{s2}G_{i1}}{G_{s4}C_i}$	$\frac{G_{s2}G_{i1}}{\left(C_{sp}s+G_{s4}\right)\left(C_i+C_{ip}\right)}$
Derivative (K_D)	$\frac{G_{d1}G_{s3}C_d}{G_{d2}G_{d3}G_{s4}}$	$\frac{G_{s3}G_{d1}}{\left(C_{sp}s^2+G_{s4}s\right)\left(C_{dp}s^2+\frac{G_{d2}G_{d3}}{C_d}\right)}$

The PID gains can be tuned by adjusting the OTA bias currents. To start, most of the summation OTA transconductance gains ($G_{s2} - G_{s4}$) can be held constant. Table I compares the PID gain terms with and without parasitic capacitances.

C. Acting

Action takes place in the robot frame. The system has control of the Pioneer robot's forward/reverse velocity and also its angular velocity. Regarding the robot arm, the arm joint angles are commanded from the control program. We used existing low level arm control routines already developed and implemented. Images of the robot Acting (grasping) a disk is found in Fig 1. Inverse kinematics are used for two joints so that the end effector has a desired height and the gripper is parallel to the ground. The height of the disk is problem specific and is hardcoded in this routine.

IV. HARDWARE IMPLEMENTATION

The next step, after successfully completing the problem in the Player/Gazebo simulation, was to try the algorithms on a real robot, Fig 1. We were able to successfully demonstrate the robot completing a two disk Towers of Hanoi problem. A Logitec model V-UBV49 Webcam was used for the camera. It was mounted to a pole on the ceiling of the lab. Fig 8 shows a comparison between the Gazebo simulation camera image and the actual image from the Logitec webcam.

There were some notable differences between the simulation and real world environments. Regarding sensing, in the simulation environment one can specify perfect illumination and ideal color values. This is not the case in a real world lab environment. In the lab one has to contend with shadows, and broader color range values. The coded range for color values had to be changed for the real world control code. The hardware also behaves differently in the simulation vs real world. The Proportional and Derivative gains (K_P, K_D) for the closed loop control system in the real hardware needed to be modified from their simulation values.

Fig 5 shows the hardware/software design flow concept for implementing an OTA based PID controller on an FPAA. Fig 5(a) shows the desired circuit. Fig 5(b) shows the equivalent Matlab Simulink model. Fig 5(c) Shows the Spice level model automatically generated from the Simulink model, Fig 5(d) shows the low level switch list for programming the FPAA, and finally, Fig 5(e) shows a picture of the utilization of the FPAA IC by plotting the switch list.

V. CONCLUSIONS

This paper presented a mobile manipulator that solves the classic Towers of Hanoi problem. The effectiveness of the Player/Gazebo simulation to real hardware design cycle was demonstrated. The process of identifying what needed to be changed to make the simulation control software work on real hardware was educational. This may lead the authors to consider during the simulation phase of a project how certain aspects of the design can be parameterized to best facilitate the transition from simulation to real hardware. Future work may consider using a camera mounted near the end effector to aid in grasping. Turn taking can be explored where the robot moves a disk and then the human moves a disk for interactive game play. Finally, the FPAA can be fully integrated into the platform for low-level control.

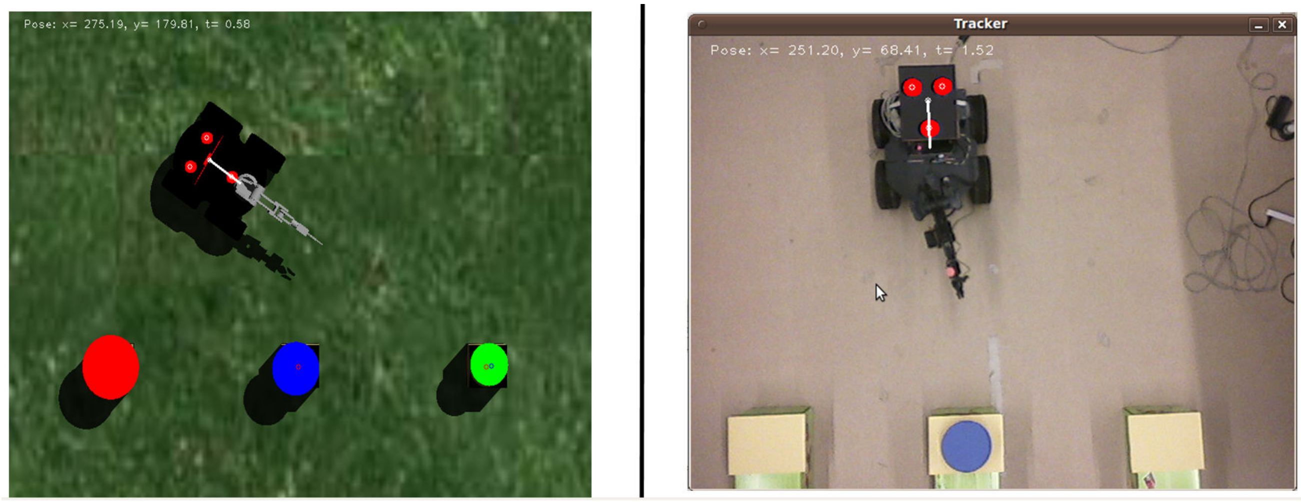

Fig. 8. This figure compares the tracker images from the overhead camera in the simulation to real life overhead camera hardware.

REFERENCES

[1] B. Gerkey, R. Vaughan, K. Stoy, A. Howard, G. Sukhatme, and M. Mataric, "Most valuable player: a robot device server for distributed control," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001.

[2] K. Stoy. (2011) The Player Project. . [Online]. Available: http://playerstage.sourceforge.net

[3] S. Koziol, C. Schlottmann, A. Basu, S. Brink, C. Petre, B. Degnan, S. Ramakrishnan, P. Hasler, and A. Balavoine, "Hardware and software infrastructure for a family of floating-gate based fpaas," IEEE International Symposium on Circuits and Systems (ISCAS) 2010, May 2010.

[4] S. Koziol and P. Hasler, "Reconfigurable analog vlsi circuits for robot path planning," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2011)*, June 2011.

[5] M. Robots. (2011, Feb.) Mobile Robots Website. [Online]. Available: www.mobilerobots.com

[6] S. Zhao, C. Chen, C. Liu, and M. Liu, "Algorithm of location of chess-robot system based on computer vision," in *Control and Decision Conference, 2008. CCDC 2008. Chinese*, july 2008, pp. 5215 –5218.

[7] A. Curtis, J. Shim, E. Gargas, A. Srinivasan, and A. Howard, "Dance dance pleo: developing a low-cost learning robotic dance therapy aid," in *Proceedings of the 10th International Conference on Interaction Design and Children*. ACM, 2011, pp. 149–152.

[8] A. Trevor, H. W. Park, A. Howard, and C. Kemp, "Playing with toys: Towards autonomous robot manipulation for therapeutic play," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 2139 –2145.

[9] M. Jones, T. Trapp, N. Jones, D. Brooks, and A. Howard, "Engaging childern with severe physical disabilities via teleoperated control of a robot piano player," in *submitted to International Conference on Interaction Design and Children (IDC) 2011*.

[10] H. W. Park and A. Howard, "Case-based reasoning for planning turn-taking strategy with a therapeutic robot playmate," in *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, sept. 2010, pp. 40 –45.

[11] E. Krotkov, "Robotics laboratory exercises," *IEEE Transactions on Education*, vol. 39, no. 1, pp. 94–97, 1996.

[12] I. J. Goodfellow, N. Koenig, M. Muja, C. Pantofaru, A. Sorokin, and L. Takayama, "Help me help you: Interfaces for personal robots," in *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, 2-5 2010, pp. 187 –188.

[13] W.-H. Fu, J. Jiang, X. Qin, T. Yi, and Z.-L. Hong, "A reconfigurable analog processor based on fpaa with coarse-grained, heterogeneous configurable analog blocks," in *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, 31 2010.

[14] P. Dong, G. Bilbro, and M. Chow, "Controlling a path-tracking unmanned ground vehicle with a field-programmable analog array," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. IEEE, pp. 1263–1268.

[15] V. Aggarwal, M. Mao, and U.-M. O'Reilly, "A self-tuning analog proportional-integral-derivative (pid) controller," in *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, 2006, pp. 12 –19.

[16] M. A. Terry, "Evolving circuits on a field programmable analog array using genetic programming," Boston, May 2005.

[17] J. Dorsey, *Continuous and discrete control systems*. McGraw Hill, 2002.

[18] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 4, pp. 559 – 576, july 2005.

[19] Y. Li, K. H. Ang, and G. Chong, "Pid control system analysis and design," *Control Systems Magazine, IEEE*, vol. 26, no. 1, pp. 32 – 41, 2006.

[20] ——, "Patents, software, and hardware for pid control: an overview and analysis of the current art," *Control Systems, IEEE*, vol. 26, no. 1, pp. 42 – 54, feb. 2006.

[21] V. Michal, C. Premont, G. Pillonet, and N. Abouchi, "Single active element pid controllers," in *Radioelektronika, 2010 20th International Conference*, 2010, pp. 1 –4.

[22] C. Erdal, "A New Current-Feedback Amplifiers (CFA) Based Proportional-Integral-Derivative (PID) Controller Realization And Calculating Optimum Parameter Tolerances," *Pakistan Journal Of Applied Sciences*, vol. 2, no. 1, pp. 56–59.

[23] A. Stoica, D. Keymeulen, M. Mojarradi, R. Zebulum, and T. Daud, "Progress in the development of field programmable analog arrays for space applications," in *Aerospace Conference, 2008 IEEE*, march 2008, pp. 1 –9.

[24] D. Keymeulen, A. Stoica, R. Zebulum, S. Katkoori, P. Fernando, H. Sankaran, M. Mojarradi, and T. Daud, "Self-reconfigurable analog array integrated circuit architecture for space applications," in *Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on*, june 2008, pp. 83 –90.

[25] S. by Willow Garage. (2010, May) OpenCV. Webpage. [Online]. Available: http://opencv.willowgarage.com/wiki/

[26] A. Singh. (2010, May) Hanoimania! Webpage. [Online]. Available: http://www.kernelthread.com/projects/hanoi/

[27] R. Borras. (2005, May) Blob Detection. . [Online]. Available: www.ros.org/doc/api/cvblobslib/html/classCBlob.html

[28] C. Erdal, A. Toker, and C. Acar, "Ota-C Based Proportional-Integral-Derivative (PID) Controller And Calculating Optimum Parameter Tolerances," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 9, no. 2, pp. 189–198.

[29] S. Liu, *Analog VLSI: Circuits and principles*. The MIT press, 2002.