Giving the Head a Hand: Constructing a Microworld to Build Relationships with Ideas in Balance Control

Arnan Sipitakiat, Department of Computer Engineering, Chiang Mai University,
Chiang Mai, 50200, Thailand, arnans@eng.cmu.ac.th
David P. Cavallo, The Media Laboratory, Massachusetts Institute of Technology,
Cambridge, MA, 02139, USA, cavallo@media.mit.edu

Abstract: This work contributes to the major promise of computational technology for learning in making discovery and acquisition of knowledge accessible in new ways and to a wider range of people. The particular focus here is on learning about ideas in Balance Control through observing one's own body motions and programming physical robots to perform balancing acts, such as balancing an inverted pendulum. The study involved two groups of learners, ages 13 to 15, over twelve months. The physical robots have a dual-mode ability that allowed learners to record and observe motions while controlling the robots manually by hand as well as under program control. A custom-made Logo programming environment together with new 2D graphical elements was created. The results not only show examples of Balance Control concepts that emerged, but highlight the learning process that was made accessible only through the computational technology.

Keywords: Balance Control, Body Motion, Logo, Robotics

Introduction: Learning by Moving the Body, Engaging the Mind, and Constructing with the Machine

Balance Control is part of our everyday life. People can walk, run, ride a bicycle, balance a stick on their finger, etc. Yet, the actual science of how we perform these tasks is extremely complex and not even completely understood. Control principles also has a reputation for being one of the harder engineering subjects (Bissell, 1999). Classical control engineering relies heavily on linear algebra, differential equations, and many other formal mathematical representations. Thus, learning balance control from this perspective is typically considered far out of reach for most learners, let alone children. It is rare for pre-university students to be introduced to this domain and even when they are, the topic is either highly simplified or presented as an introduction for further studies. See (Kolberg, Reich, & Levin, 2003) and (Miller, 2001) for examples.

On one hand, this work presents a creative method to make some Balance Control ideas more accessible to younger learners. But more importantly, it demonstrates how computational technology can provide new fertile ground for the acquisition of scientific knowledge by learners. Rather than looking at computers and digital media as just another platform for information delivery, the approach adopted here emphasizes the role of computational media in forming, expressing, testing, and debugging one's own hypotheses about phenomena at hand.

We have utilized the fact that children can perform many balancing actions naturally as a starting point into a scientific investigation and a rich learning process. Two balancing tasks, an inverted pendulum; and a balancing beam, were created where learners can carry out the balancing tasks manually by hand and compare their motions against the behavior of a robot that they program. This dual-mode ability has been augmented by applying ideas in Computerized Dynamic Posturography (CDP), a field that utilizes digital sensors to record and study how humans maintain balance during a gait cycle (Nashner, Black, & Wall, 1982). Together with the use of digital sensors, a custom-made Logo programming environment has been created allowing learners to manipulate the recorded data (e.g. to playback their motions at faster or slower speeds, to calculate rate or speed of the falling pendulum, or to plot a graph of two or more variables). A new 2D graphical component has also been introduced to support the learners in creating visual cues and representations that assist them in the learning process. Initial results demonstrate encouraging examples of how the digital medium has created new ways for children to construct their understanding of some ideas in Balance Control through active observation, creating hypotheses, and testing through programming robots.

The Designed Activities

We have applied and simplified CDP techniques by focusing on sensing motion of robotic mechanisms (not the human body itself) that can be controlled by the body. A special requirement for any designed activity was that it must allow the learners to accomplish the task both manually by hand and autonomously under program control. The hardware configurations of both modes must be identical to allow for their motions to be compared against each other. Given these requirements, two activities were initially developed for the learners.

The Inverted Pendulum (The Cart-Pole Challenge)

The setup was an inverted pendulum mounted onto a robotic vehicle (Figure 1 left). The challenge was to prevent the pendulum from falling. Two versions with the same physical properties were built. One had no motors and was used to record the learners' motions balancing the pendulum by hand (pushing the car back and forth on a track). The second was equipped with two DC motors and was controlled through the Logo programming environment. Note that the motions of the motorized pendulum can also be recorded.

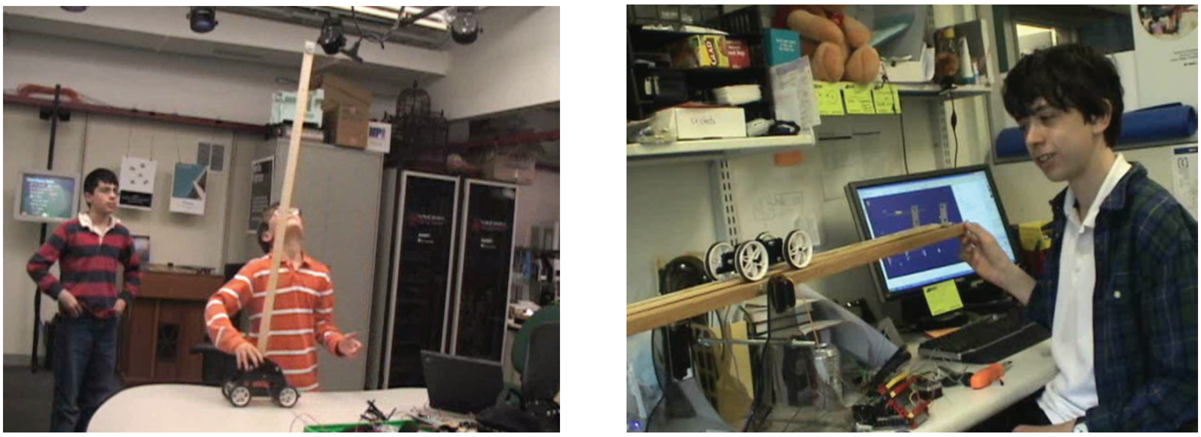


Figure 1. Students playing with the inverted pendulum (Left). The goal was to prevent the pendulum from falling. The balance beam (Right) being manipulated by a student. The goal of this challenge was to prevent the rolling car from falling off the beam.

The Balance Beam

The goal of this task was to balance a robotic vehicle that could travel back and forth on a pivoting beam (Figure 1, right). When the car was pushed away from the center, the learner had to control the beam to prevent the vehicle from falling off and also return it back to the beam's center. Only one setup was built to serve both the required operational modes. The beam was motorized but the motor used was large enough to tolerate the reverse torque that was applied when the beam was controlled by hand. The vehicle's position on the beam and the beam's angle were sensed.

Physio-Syntonicity: A Computational Means to Make Balance Control More Accessible to Children

Programming was the primary expressive means for the learners. A special version of the Logo programming language called PyoLogo (Python-Open-Logo) was created. PyoLogo included added functionalities such as the ability to record and playback robot motions. Figure 2 shows a screenshot of the programming Environment. Logo was picked because of its well-established learning philosophy underlying its design. The epistemological goal of connecting robot balance to human balance was that it could transform a topic previously alien and unknown (control engineering) into a something more familiar and less intimidating (one's own body actions). This approach resonated well to Logo's popular turtle metaphor that allowed a learner to relate mathematical operations (geometry) to one's own actions moving in a physical space. This connectedness or body-syntonic quality has been the main focus of Logo and their visionaries such as Seymour Papert since the early days of using programming as a learning tool for children (Papert, 1980).

In the same spirit as Logo's turtle, our work with students has led us to develop a computational means that allowed a number of previously abstract control techniques to become more accessible to younger learners. The case-studies will show how familiar properties of 2D on-screen objects were used to perform and analyze balancing actions. We named this approach the Spatial Computing Paradigm (SCP). The combination of SCP and using body motion has led to a framework for learning that is "physio-syntonic". While body-syntonicity, as mediated through Logo's turtle, has connected geometry to familiar body movement in space, physio-syntonicity has opened up new pathways for children to access ideas in Balance Control through familiar actions both on and off the computer screen.

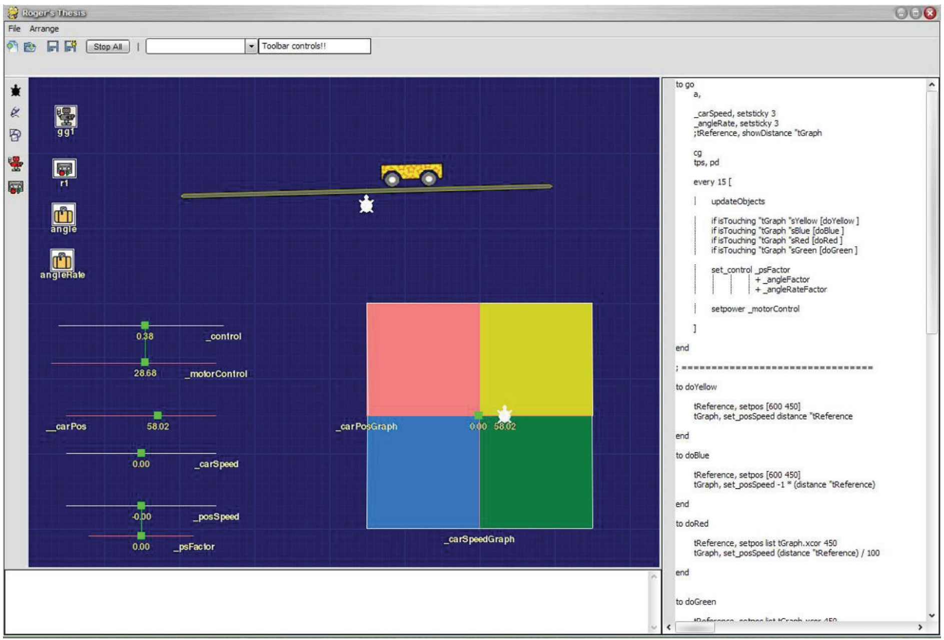


Figure 2. A screenshot of the PyoLogo programming environment

There have been examples of using people's experiential knowledge as a basis for the learning and teaching of ideas in balance control. In the 1980s, balancing a bicycle was used as part of the control course at the Mechanical Engineering Department at the University of Illinois (Klein, 1989). Students were challenged to explain how a bicycle works. Different aspects of the bicycle were analyzed by custom building bicycles with rearranged characteristics. This approach was adopted by many other universities and newer iterations incorporated sensing devices that allow for computerized analysis (Astrom, 2005). These approaches, however, focused more on connecting experiential knowledge to the traditional approaches towards control. This work pays more attention to designing new learning trajectories and new computational approaches towards control that could be both practical and accessible by younger learners.

Research Methodology

We have conducted a design research. That is, we were not testing a finalized pre-constructed learning environment with children. Quite the contrary, we started with only a rough implementation of the tools. The direction of how the activities were to proceed was also not fixed. It was the experience of working with students that advanced the researchers' thinking and helped refine the tools to better support the activities. Thus, the research methodology needed to reflect this design nature where innovations come not only from the researcher but also from the learner and the interactions between the two.

Much of the process in this work resembles the well known participatory design approach (Schuler, 1993) where the researcher's inquiries with students were highly contextualized and the students (or users) played a significant role in determining what aspects of the tools worked and what did not. But it has also put an emphasis on applied epistemological anthropology (Cavallo, 2000) as a means to identify learner's common ways of thinking about the phenomenon. It then directs the design of the environment to make use of these ways of thinking in the learning process.

Participants

Two groups of students were involved in this research. All are volunteers from a local school in Cambridge, MA, USA. The first group consisted of Albert, 12, and Anderson, 13. Both students participated in eight sessions throughout a seven week period. The second group consisted of Greg, 15 and Joe, 14. Joe participated for the first six weeks (seven sessions) before he had to move to another country. Greg was the longest participant with twenty four sessions throughout a twelve-month period. Three out of the four students had no prior programming experiences. Greg had done some work with Logo. However, all of them used a computer on a daily basis for general purposes such as e-mail, instant messaging, web browsing, gaming, etc.

Some Results

Evolution of the Approach: Negotiating Meaning When the Body Meets the Mind

This section describes the evolution of the programming environment. The researchers had made initial assumptions about possible approaches and means that could allow the learners to accomplish the balancing tasks. But early interactions with learners had revealed unforeseeable aspects of the learners' thinking that was not necessarily compatible with our assumptions. The tools were then adapted to better accommodate both the learners way of thinking and what was needed to produce a functional system. A summary of the phases in the evolution is illustrated in Figure 3.

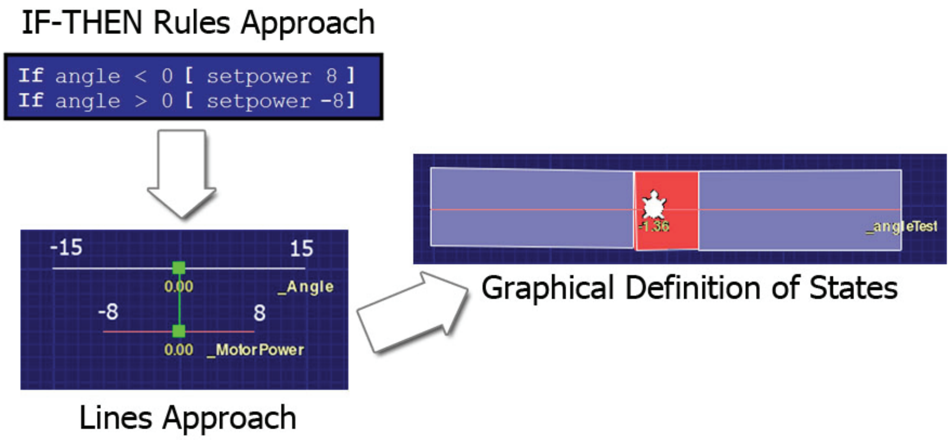
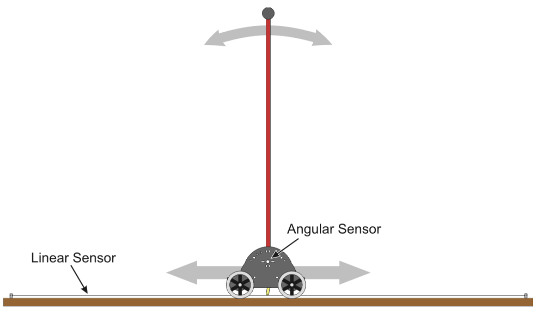


Figure 3. The representation of "state" evolved from simple IF-THEN rules (top-left) to the researcher's Lines approach (lower-left). The graphical plane approach (right) was a result of the researcher's observation of the students' thinking combined with the researcher's knowledge about what was needed.

Phase I: The IF-THEN Rules Approach

When the students were asked to observe and explain how they believe they were performing the balancing actions, we quickly noticed that all four students described the process in terms of discrete states; "If this happens, then do that." Despite being rather inaccurate, all their explanations were expressed in this form. For example, when we first worked on the inverted pendulum challenge, both groups came up with the IF-THEN rules as shown in Figure 4.



Description of the strategy:

```
If beam_falls_to_the_left
   Then move_the_car_to_the_left
If beam_falls_to_the_right
   Then move_the_car_to_the_right
```

Implementation in Logo:

```
If angle < 0 [ setpower 8]
If angle > 0 [ setpower -8]
```

Figure 4. (Left) The inverted pendulum setup. (Upper right) Learners' initial description of how to control the car to balance the inverted pendulum. (Lower right) The implementation in Logo. "Angle" refers to the inverted pendulum's angular position where zero is the upright position. "Setpower" determines the motor speed that drives the car. The numbers 8 and -8 represent "full motor power" in each direction.

Using IF-THEN rules allowed the students to quickly construct a program that matches their thinking. Despite its simplicity, there were two problems. First, this simple idea was not sufficient to keep the inverted pendulum from falling. Although there were many ways to explain what was missing, thinking about "anticipation" was the approach used in this work. Generally speaking, during every move, the robotic car needed to slowdown even before the pendulum returned to its upright position and let its momentum of inertia do the rest of the work. That is, the car needed to anticipate the affect of momentum of inertia.

Learners' immature reasoning was, of course, expected as part of the learning process. However, the IF-THEN rules approach did not serve well as a tool to think with. When rules become more complicated, tuning the numbers became tedious, and it quickly became impossible to observe how the rules were affecting the behavior of the robot. These limitations soon became an impasse leading to a need for something different. On one hand, there have been examples of solutions to the limitations of textual expressions such as the "live text" in FLogo (Hancock, 2003). However, because a significant motivation of this work had to do with visual observations of physical objects, we did not want to focus only on text.

Phase II: The Lines Approach

"Lines" is an approach developed prior to the fieldwork by the researchers. It mainly provided a means to handle scaling operations. Figure 5 shows how to implement Phase I's IF-THEN rules. An angle value can be converted into a motor power level with a pair of lines. The main advantages were that the operation can be visually observed and tuned (by stretching or moving the lines). It also allowed for proportional control. As opposed to either moving full speed one way or the other, the motor power can be conveniently configured to scale with the pendulum's angle.

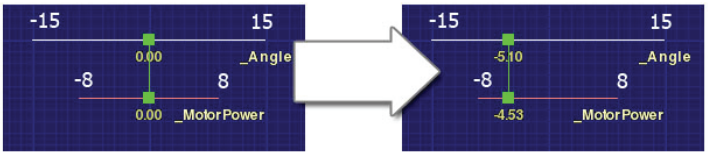


Figure 5. An example of how a "lines" approach can convert an input value (angle) into a proportional output value (motor power). As the inverted pendulum falls from the upright position (from Angle = 0 degrees to Angle = -5.10 degrees), the motor power value changes as well (from 0 to -4.53).

Although this method could produce better results, we observed that it also did not work well as a thinking tool for the learners. It became particularly clear when the program became more complex where multiple pairs of lines were used, as shown in Figure 6. We ended up with similar problems found with the IF-THEN rules approach in terms of poor comprehensibility of what was going on during program execution. As a result, one group quickly went back to the more graspable IF-THEN rules approach while the other became stuck, which was a setback to the development of their thinking.

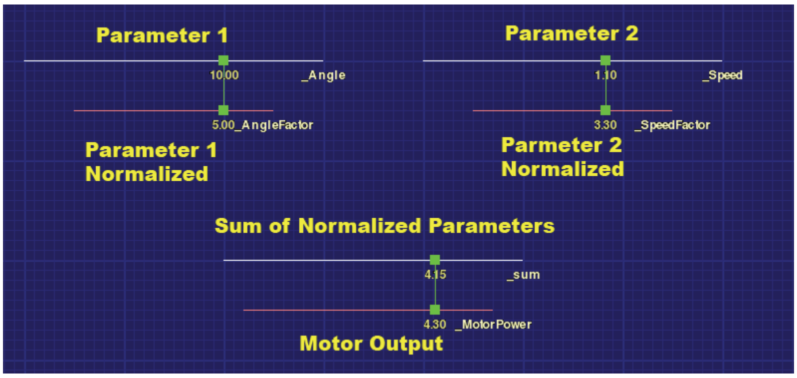


Figure 6. An example of how confusing the line approach became when multiple pairs were introduced. Although it could generate good results, it failed in terms of human comprehension. Many values would be moving on the screen while the learners had little understanding of how they were affecting the robot.

Phase III: A New Graphical Approach towards Defining States

After examining the limitations of phase I and II, Phase III was essentially the result of a negotiation between what was needed and the observations we had made of the learners' way of thinking. This new approach was inspired by the learners' drawings of their thinking on paper. They often divide a variable under investigation into regions that can be visually observed (Figure 7, left). Thus, we started to experiment drawing rectangular areas to define regions on a line object. This technique allowed students to easily resize and, thus, modify the state boundaries (as shown in Figure 3, right). Later, the idea was expanded to work with two variables. It allowed a visual mapping of the four possible variable combinations (Figure 7, right). A turtle was programmed to move depending on the input values. The current state was determined based on which rectangle the turtle was touching. Learners can then program the robot to behave differently depending on these states. In essence, it was an approach that made defining states both practical and comprehensible.

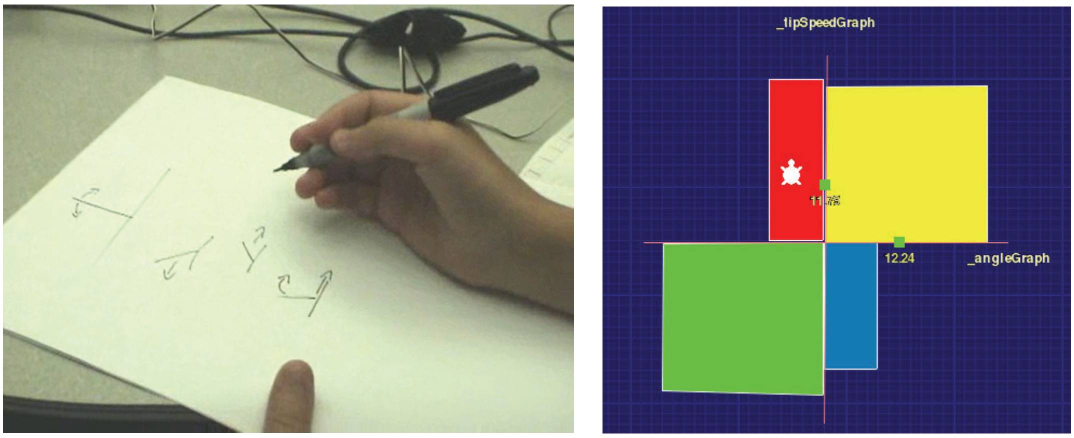


Figure 7. (Left) Observations of how learners often explain their thinking by drawing different regions for a variable under consideration. (Right) Using shapes to define regions of states. The figure shows the pendulum's angle on the horizontal axis and the top-tip's speed on the vertical axis. The turtle's location depended on these two variables. The robot can be programmed to act differently depending on the rectangle over which the turtle was located.

Observing the Body: Learning by Reflecting on Recorded Motions

While the tool was evolving, the development of the learners' strategy was developing as well. In this latter process, we have found that observing body motions while performing the desired actions have provided a rich resource for learners to develop ideas that they would then pursue.

There were two main advantages of observing recorded motions. One was that it has led to rich discussions about the current strategy being explored. The learners can evaluate their results while the researchers could add their perspectives and introduce new ideas, such as the role of speed (or rate). Adding a speed parameter is a simple way to introduce anticipation to a balance control model. But learners' understanding of this concept did not take place on its own. As mentioned earlier that all learners incorrectly believed that the car's movement should be proportional only to the angle of the falling pendulum. The role of speed was not recognized. In fact, all four learners had falsely confirmed that their body was following their (inaccurate) description! Only after reviewing the recorded data and doing some analysis (e.g. graphing) that the learners became more acceptable to the aspects that defy their current thinking. Overcoming one's own thinking is not always a simple process, as seen in Imere Lakatos's influential book "Proofs and Refutations". See (Lakatos, 1976). This was where the second benefit of observing recorded motions came in.

When the learners were in need of new ideas and directions to continue, observing recordings of a working model was useful as an idea generator. The learners would notice some aspects of the movements and develop an idea to pursue and move on forward. However, accepting ideas from the learners implies that the researchers must allow learners to develop ideas that can differ, often significantly, from conventional control approaches. We believe this aspect is extremely important for a design research, as it provides new insights into how control situations may be interpreted and approached by young learners.

For example, when Albert and Anderson started to realize the importance of anticipation, the researchers tried to explain to them the role of angular rate, which is a typical approach utilized with an inverted pendulum. However, Albert and Anderson had difficulties making sense of what angular rate was and how it was useful. Both learners tried to experiment with it but became even more confused. During that time, Albert had noticed that the top-tip of the pendulum was the part that moved the least compared to the other parts of the inverted pendulum. He then hypothesized that this tip might actually be the main focus and that its movement needed to be minimized. He asked if it was possible to use the top-tip's speed instead of the angular rate in his model. Posed with this idea, even the researchers were unsure about its validity. But we decided to go ahead with it because it was an approach that made more sense to the learners. From a learner's perspective, a mysterious "angular rate" was substituted by a more straightforward "tip speed".

The tip speed idea eventually turned out to be inferior to the angular rate. But although it did not allow for a working system, its main contribution was in the learning process. Because Albert and Anderson owned the idea and understood well what the variables were, both students were able to spend a great deal of time experimenting how speed affected their model. Thus, the concept of speed became less alienating. It was much easier to introduce other kinds of speed variables to them during later iterations.

It really is Rocket Science: Rediscovering a Space Shuttle Technology

Once the learners agreed in principle that a second variable (speed) was needed in the control model, the next question was how to implement it in their program. We have learned that IF-THEN rules and the lines approach had serious limitations. This was when the development of the graphical representation of states (as described earlier) was created. For the inverted pendulum project, the learners (with the help of the researchers) created a 2D space where the horizontal axis represented the pendulum's angle and the vertical axis was the pendulum's tip speed (Figure 7, right). This arrangement resulted in a combination of four states (positive position with positive speed, negative position with positive speed, and so on). The learners were able to draw rectangular shapes in these regions to specify how the robot should react. A turtle was programmed to move in this 2D space indicating the current angle/speed combination. It gave the students a comprehensible view of what was going on during program execution. This method allowed the inverted pendulum to stay up-right for a short time (roughly eight seconds) but it worked extremely well for the second project, the balance beam. The beam was able to successfully return the car back to its center.

The researchers later discovered that this latest approach was significantly similar to a well-established engineering method called Phase-Plane Logic Control (Wie, 1998). Phase-Planes have been most notably used in the space shuttle orbital control in the early 1980s as shown in Figure 8. This numerical method utilizes a graphical means that defines regions on a 2D space that are remarkably similar to the regions created by the learners. The main difference was that Phase-Planes are typically translated and implemented as look-up tables while our approach utilized the actual graphical components themselves to carryout the operation.

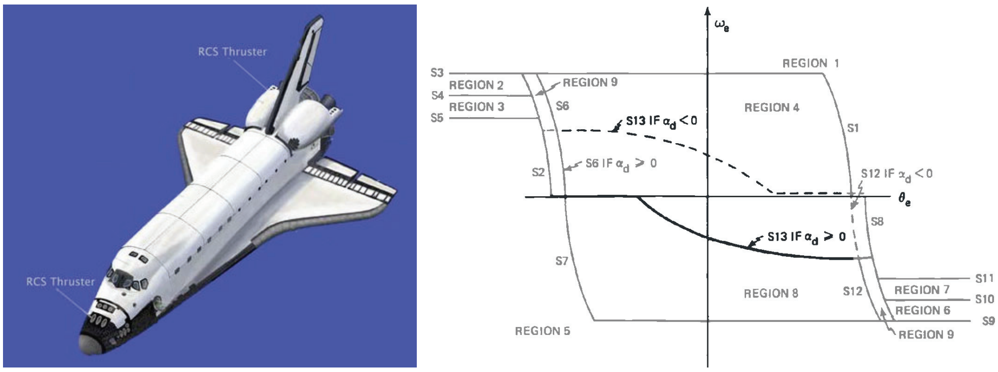


Figure 8. (Left) the space shuttle's on-orbit reaction control system (RCS) utilizes multiple thrusters distributed throughout the head and tail sections. They are controlled through a Phase-Plane diagram (right) which was used to determine when to fire the appropriate thrusters

Conclusions

By connecting body motion to Balance Control, we do not imply that learners would discover the underlying principle purely through observation. We have showed that using body motion connects the learning activity to something familiar and personal. Programming a robot to perform a balancing task that learners can perform manually with their body creates a context, stimulates curiosity, and provides motivation for learners to reflect on their body movement and create hypotheses. The computational tools then allow learners to express their ideas in programmatic terms, test them, debug them, and evaluate them. For a teacher, these concrete expressions allow for better understanding of the learner's thinking, which helps the teacher to provide new ideas and insights in ways that fit the current situation and are meaningful to the learner.

The discovery of the Phase-Plane Control Logic has provided good evidence that the design process was both valid and functional. It would have been difficult, if not impossible, for anyone to foresee that a Phase-Plane Controller would work well with children without spending time learning about how children think. The implementation of these representational forms and methods has a long way to go. There is no doubt that much more can be done and many other Balance Control topics can be introduced to learners by expanding the methods presented in this work. But whatever the possibilities are, involving students in the design process will remain the key aspect.

The reason we should care about such new representational forms is because of the quality they have that promotes learning in ways that cannot be done with paper and pencil. For example, the graphical component of the Phase-Plane approach allowed the method that resonated with learners' thinking to become practical by doing it repeatedly and accurately. Traditional abstract mathematical methods such as lookup tables, though would have worked, would have taken away the graspable quality that has been key to the success of this research. It is this characteristic that makes computational forms worth paying serious attention. It opens up new learning pathways that highlight human comprehension.

References

Arbib, M. A. (1969). *Theories of Abstract Automata* (1st ed.). Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Astrom, K. J., Klein, R.E, Lennartsson, A. . (2005). Bicycle dynamics and control: adapted bicycles for education and research. *Control Systems Magazine, IEEE, 25,* 26-47.

Bissell, C. C. (1999, October 1999). Control Education: Time for Radical Change? *IEEE Control Systems, 19,* 44-49.

Cavallo, D. (2000). Emergent Design and learning environments: Building on indigenous knowledge. *IBM Systems Journal, 39*(3 & 4), 768-781.

Hancock, C. M. (2003). *Real-time programming and the big ideas of computational literacy.* Unpublished Thesis Ph. D. --Massachusetts Institute of Technology School of Architecture and Planning Program in Media Arts and Sciences 2003.

Klein, R. E. (1989). Using Bicycles to Teach System Dynamics. *Control Systems Magazine, IEEE 9,* 4-9.

Kolberg, E., Reich, Y., & Levin, L. (2003). Project-based High School Mechatronics Course. *Internation Journal of Engineering Education, 19*(4), 557-562.

Lakatos, I. (1976). *Proofs and refutations: the logic of mathematical discovery*. Cambridge; New York: Cambridge University Press.

Miller, D. P., Stein, C. . (2001). Creating autonomous roboticists. *Intelligent Systems, 16*(2), 20-23.

Nashner, L. M., Black, F. O., & Wall, C. d. (1982). Adaptation to altered support and visual conditions during stance: patients with vestibular deficits. *J. Neurosci., 2*(5), 536-544.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Schuler, D., Namioka, A. (Ed.). (1993). *Participatory Design: Principles and Practices*: Lawrence Erlbaum Associates.

Wie, B. (1998). *Space vehicle dynamics and control*. Reston, VA: American Institute of Aeronautics and Astronautics.