PROCESS FOR CREATION OF SUSTAINABLE WEB 2.0 BASED INTERACTIVE DEMONSTRATION AND CONTENT FOR CONTROL EDUCATION

by

JOSHUA C. UPDYKE

B.S., Kansas State University, 2008

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Mechanical and Nuclear Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2011

Approved by:

Major Professor
Warren N. White

Abstract

The objective of this research was to show that controls demonstrations and education can be greatly improved with the use of Web 2.0 tools. First, a user-centered and collaborative website was developed using the freely available Web 2.0 tools. Then procedures for creating animations using control system simulations were developed. Using the Java programming language, interactive graphs were constructed for display to allow the viewer to change the initial conditions of the controls system and plot the results of the simulation as calculated. These animations and interactive graphs used already developed MatLab resources.

Using Web 2.0 tools, the collaborative web site can be easily updated by several people simultaneously, only requiring access to the internet. Because of these advantages, collaboration between multiple universities and controls researchers in distant locations is possible, and the new website has the potential of becoming the center of controls research around the world. More importantly, the procedure allows future systems to be demonstrated while requiring minimal additional work, thereby increasing the universities' ability to educate their students, as well as the general public, about controls system research.

Table of Contents

List of Figures

Dedication

I would like to dedicate this work to my loving parents. Without their knowledge, wisdom, guidance and unconditional support, I would not be able to reach for my dreams.

I am honored to have you as my parents.

Chapter 1

Introduction

In this thesis, we present a method to display and help educate students about controls systems and controls research. One example will illustrate the need and motivation. At Kansas State University it has been difficult to demonstrate the ongoing research projects. Several professors and their respective students are working on various research in controls. Because this research is segmented across several different people, there has been no common location to display their work other than the traditional publishing avenues, and it has been difficult to call attention to the collective effort. No standardized process existed to make their work available to the public.

This problem is not unique to Kansas State University. Many universities and organizations are performing interesting work in this field. However, it is difficult to draw attention to this research for lack of a venue. Many universities, including The University of Michigan and Indiana State University, have a very small website presence for controls research, if any at all. This creates several problems. First, this research has the potential to be used as teaching aids. It is impracticable for an entire class to visit one of the research labs or see a demonstration of the controls system. Additionally, while it might be possible for a single professor to create teaching aids such as videos or handouts explaining the topic, this is unlikely. Because the material to create these kinds of demonstrations is maintained by different people, it becomes tedious to transfer the files to the person who needs them. Currently, this is done by one professor asking another for the material. This system hinges on a previous knowledge of what materials the other professors have. Most research is hard to find without prior knowledge of where to look, and it is unlikely any one professor would know about all the opportunities available for teaching aids. A common location for sharing educational materials does not exist. Due to distance and different countries and languages, research being done at other universities and organizations may be more difficult to use because of an inability to meet in person.

Drawing attention to ongoing research has several benefits to the people doing the work, as well as benefits to the department and the university as a whole. For research which is already funded, the added visibility serves both the funding source and the research. For unfunded research, journal papers are the main avenue in the proposal process. However, not all research is at a point where papers can be written. By adding this research to the website, it is possible to start building interest in the topic. Students looking for research opportunities or other researchers with similar interests will have a way of learning about the current progress. As the research progresses, the topic could gain enough interest to improve the chances of a successful research proposal. Aside from grants, an increase in attention can result in other funding, increases in student enrollment, and creating new research opportunities. In general, more visibility can promote greater opportunities.

1.1 Motivation

At the beginning of this project there was no central location to display the research preformed in the Mechanical and Nuclear Engineering Department at Kansas State University. The department had a simple website, maintained by a single person. While simple changes to the website were regular, it was difficult to add new content in the research areas. All changes to the website needed to go through the website administrator. This meant that the people doing the research could not update the website directly, but instead had to ask the website administrator to make the changes on their behalf. Specifically, when new content relating to non-linear controls research was added, pages were created by a graduate student and then passed to the website administrator to be added. Once added, several links did not work correctly, files were labeled incorrectly, and videos could not be viewed. The method for adding content to the website was slow, difficult, and lacked collaboration.

The goal of this project was to create a new website which solves these problem. The new website needed to be easy and quick to update, so it could be updated often. The website needed to be collaborative, so many groups could add content about their respective research topics. A standardized method for updating the website needed to be created, so that future contributors could easily add their work. Finally, the website needed to take advantage of visually impressive methods for demonstrating the research.

1.2 Preview

Chapter two will discus the current state of controls media and the problems associated with it. The lack of collaboration, infrequent updates and issues related to making video content available are addressed. Next, the advantage of using Web 2.0 tools is covered in chapter three. The creation of the new controls website for Kansas State University is used to show how these tools can solve the previously mentions problems. Chapter four covers how YouTube can be used to simplify displaying video content. Chapters five and six detail the motivation and creation of higher level content to replace traditional graphs. Finally, the resulting benefits are restated in the conclusion.

Chapter 2

Current State of Controls Media

In recent years the Internet has become one of the primary sources for information. It holds vast amounts of information across millions of websites. However, most of this information lacks any uniformity and is segmented across many websites, most of which are not well organized, maintained, and often contain out-dated information. Not to mention the sites which contain inaccurate or only a portion of the topic material. While the Internet offers an abundance of information, actually making use of this tool can be a daunting task.

Traditional websites are being replaced with Web 2.0 versions, which offer additional features and are more usable. However, many websites are slow to adopt these new practices. The controls field is one such area that has yet to embrace the possibilities of Web 2.0. Many universities as a whole have adopted some of the practices; however, the specific departments researching controls are not taking advantage of these tools, and are still relying on traditional website methods.

2.1 Traditional Websites and Associated Short-Comings

The majority of websites on the Internet do not use Web 2.0 features. This means that the websites are usually written in HTML and created with some website creation tool or programmed by hand. Then, these files are uploaded to a server using some file transfer software. This method has been the standard for most of the history of the Internet. This leads to several problems, specifically problems in updating and lack of website collaboration.

2.1.1 Infrequently Updated Websites

"According to recent studies, one in three firms admit that updates are being done infrequently."[1] Streamline.nets latest Small Business Bytes Survey finds that many small companies with a website struggle to maintain a compelling web presence. "Despite two-thirds having had a business website for two years or more, only one in ten firms succeed in making updates on a daily basis. One in four companies admit to updating their websites on a monthly basis. Over half (fifty-four percent) admit to making technical tweaks only infrequently."[1]

A key aspect of designing a website for education and demonstration of controls material is the ability to update it easily and often. In order for a website to be a useful tool, it must provide content in a timely manner. If the content is available through other means before it is available on the website, then those means will become the dominant method. An infrequently updated educational website will not be visited often. Websites that add content regularly attract visitors who return regularly.

Websites are updated infrequently because of two main reasons. First, it requires a lot of time and money to update a traditional website. Specialized knowledge, training, and tools are often required to update traditional websites. Many organizations hire an outside company to develop and maintain their websites for these reasons. However, organizations that are unable to afford the expenses create and maintain their own websites and assign a current employee the task of maintaining the website on top of his or her previous responsibilities. This method usually results in a minimal website with limited features.

Secondly, the person who is maintaining the website is rarely the person who has the new content to add. This makes it more difficult for new content to be added because it must first be passed from one person to another. If the website updating responsibility is limited to a few, or even a single person, the website will most likely not be updated often. Additionally, if the person in charge of updating the website has other responsibilities, website updates may not be a priority for him or her. This intermediate step can cause long delays for new content to become available to the public. Because the content being added may be of a specialized area of research, mistakes can be made. If the person updating the website does not fully understand the content, information may become transposed or incorrectly labeled. In general, the person with a vested interest in changing the website will do a better job.

2.1.2 Lack of Collaboration

Because many websites are maintained by a single person or small group, the website lacks collaboration. At universities many groups have valuable content to add but lack the technical skills or resources to host a website by themselves. Traditional websites do not offer a method for the people who visit their site to add any information. It is common for visitors to have a small amount of information they would like to add-perhaps a specific example or a paragraph relating to a topic already covered on the website. These visitors do not have enough content to require a full website of their own, or even an entire page. However, when you combine all the small amounts of information from many visitors, traditional websites are not taking advantage of a huge resource.

Every small unit of contribution is important to a Web 2.0 service. Millions of such contributions eventually lead the website to a state of higher relevance. As ”For instance, any conventional media company (employing hundreds of reporters) has today been easily beaten by blogging platforms like Blogger and WordPress in producing extremely frequent and relevant content as millions of users are acting as a contributor, building up a large resource within a much lesser span of time.”[11] Prashant Sharma has noted.

2.1.3 Video and File Issues

Various files are used to demonstrate and educate about control systems. The KSU Mechanical and Nuclear Engineering Department’s website had video files, pdf documents and several MatLab and Simulink files. However, it was difficult to organize these files on a traditional website. The files were spread over several pages with little or no description about what the files were. Video files can be a particular problem. For example, the video files on the Mechanical and Nuclear Engineering website were previously large files which required a long time to download. Once downloaded they required a specialized media player (RealPlayer) in order to be viewed. With the proper encoding these videos could be played directly from the Internet, or at least put into a more compact format for easier downloading and viewing with any number of media players.

2.2 Additional Website Example

The controls area of Kansas State University is not the only website that suffers from these issues. Several universities have sections of their websites related to the education and demonstration of controls topics. One such university is Carnegie Mellon, the website of which focuses on tutorials for controls using MatLab and Simulink. Carnegie organizes the topics into a logical progression for students but relies on traditional website tools. Each page contains text and images which detail the topic covered. However, the information is static. The viewer has no way to interact with the examples, cannot ask questions, or contribute to the website. While the website contains valuable information, no new content is being added and apparently no mechanism is in place to expand the content covered.

2.3 Summary

Traditional websites can be problematic for displaying educational material and research. Maintaining a website of this kind requires a lot of time and can become a dedicated job. However, it is often not an option to hire a professional, and it is very unlikely that professor will have the time or expertise to handle this task. Another solution needs to be found that can allow educators and researchers to focus on their primary work. Web 2.0 tools offer one such solution.

Chapter 3

Web 2.0

”The term Web 2.0 is commonly associated with web applications that facilitate interactive information sharing, interoperability, user-centered design, and collaboration on the World Wide Web.”[11] The most familiar example of this is Wikipedia. This website has been able to accomplish a very difficult task by using Web 2.0 techniques. Currently, the top ten most visited websites all use Web 2.0 features.[8]

3.1 Web Based Platform

A major drawback in traditional website development is the need for software development packages installed on a local machine. Most websites are created locally, as a private file, and then uploaded to a server to be viewed publicly. Web 2.0 sites use the Internet as the development platform. This removes the dependency on operating systems and development packages. Whatever method of Internet access (Windows, Mac or Mobile), the website can be updated. By making the method of updating the website more accessible, it is more likely it will be updated.

There are many methods of doing this. However, we choose to use Tiki Wiki[7], a free web-based application for maintaining websites. By building the controls website on a wiki platform, we are able to update the website much easier. To update the website, users can log onto the website from any web browser, anywhere in the world, as long as the users have access to the Internet. Once logged in, the users can edit the content of any page almost instantly. The editing tools are very similar to other word processing software such as Microsoft Word. By using a familiar style of editing, the learning curve of the website creation is reduced and the requirement of learning difficult programming languages like HTML is removed. The content displayed during editing appears very similar to the final output. When the users are finished making changes, they simply save the changes to the page, and it is immediately updated for others to see.

3.2 Collaboration

"Wikipedia takes first place when it comes to proving the power of collaboration. Before 2001 (year of Wikipedia's inception), there used to exist only driven information sources such as Encyclopedia Britannica, About.com, and other similar sources where collaboration was never implemented. Today, Wikipedia stands way ahead in terms of content quantity as well as quality".[11]

By including the editing tools with the website, it can be edited by anyone with a user name and password. This means that multiple people can be updating the website simultaneously. The level of collaboration can be controlled by the distribution of user names. User names can be given to anyone who requests one or restricted to a small group of people.

In addition to deciding who is able to receive a user name, users can be given varying levels of control over the website. Privileges which allow users to perform specific tasks can be assigned. Examples include changing website pages without approval, viewing private pages, and uploading files. Groups can be made to define a set of privileges for several members. This makes it very simple to maintain control while having many people collaborating on a single website.

3.3 Interaction with Viewers

The new website includes tools used to create and maintain Internet forums, on-line discussion sites where people can hold conversations in the form of posted messages. This allows website viewers to leave feedback on the site, as well as communicate with other viewers. A forum takes advantage of the community of people interested in the web site's topic as a resource for adding content, answering questions and generating new ideas.

Forums are currently being used for nearly every topic. Open-source projects often use them as a way for people around the world to collaborate on a single project. An example of this is the low-cost open-source rapid prototyping system, RepRap.[3] While initially the project was developed by Adrian Bowyer at the University of Bath, it has been greatly contributed to by the forum members. The forum contains over 50,000 messages from thousands of contributors around the world. These messages include developmental discussion and questions being answered by peers.

Viewers being able to easily interact with a website opens up a huge potential resource for demonstration and education. Viewers are able to ask questions similar to those in a classroom environment. These questions can be answered by an administrator of the website but, more importantly, can be answered by other contributors of the website. Consequently, this builds a community of people interested in a common topic who are available to help each other. Additionally, questions that have been answered remain on the website for new viewers to read. The questions and answers become content of the website.

Using the viewers as a resource for improving the website allows it to grow at a much faster rate than would be possible with other methods. In general, as the viewers of the website increase, the website improves and, in turn, attracts more viewers to the website. Allowing the viewers / contributors to interact with each other offsets the work load for one person or group maintaining and updating the website. The community updates and maintains the website instead of the single person or group.

3.4 File Repository

The new website allows a file repository to be created. This makes it much simpler to maintain a database of files. The wiki will create a familiar file directory for organization of files. This tool is customizable to control who has the ability to add and remove files. This makes it much simpler to share files with many people on different computer networks. Additionally, files can be uploaded and stored on a website without the need of specialized software for uploading files. This makes it possible for the community of users to help add content, add descriptions, and maintain the current file collection.

3.5 Maintainability

A website must be maintainable for future development. At universities the people involved in a project are constantly changing. Students graduate and new students take their place. Because the development of the website is decentralized, a Web 2.0 site is less prone to becoming stagnant. Graduate students can create and edit sections of the website pertaining to their individual work. As students leave, new students will take over updating the website as they continue or start new research.

3.6 Summary

Web 2.0 tools add much needed functionality to website design and maintainability. Because it is easier to add content and for many people to work together, a website, using these tools, is more likely to be updated often. The additional tools such as blogs, forums, and file repositories offer new methods to interact with viewers, which make the website more useful. The underlying goal of websites is to make information available. Web 2.0 tools make this easier and faster.

Chapter 4

Popular Media Methods - YouTube

Engineers often avoid reinventing the wheel by building on previous work. However, when it comes to demonstrating and educating about controls topics, this is not always the case. A great example of this is video hosting. The Mechanical and Nuclear Engineering Department of Kansas State University has several impressive videos from various projects. Almost none of these are displayed on the university's website, or anywhere else. These videos, which could draw attention, prospective students, and visibility to the department, are underexposed. Neither are they being used as teaching aids. This chapter will address the reasons for this, and a solution through taking advantage of a free resource.

4.1 Causes of Underexposed Videos

The first reason that these videos are not being seen is lack of a central location for repository. While the department has a public website, it does not have a dedicated section for video content. Many professors are not aware of the opportunity to publicize their videos, nor are they aware of the possible benefits for doing so.

The second reason is how difficult it is to make these videos available. Making high quality videos available on the Internet can be problematic. Videos are usually large files compared to other Internet content. This makes the videos slow to load from websites with limited bandwidth. If the web servers are not set up to handle this load, they may even crash. Because of this problem, less impressive content requiring less bandwidth has been given preference. In addition to this, videos can be stored in a variety of formats, each of which has different requirements for playing the videos. Because these videos are being created by different groups with different tools, and different situations, trying to organize all the videos into a standard format would be a difficult and time-consuming task. Additionally, all the problematic issues with updating a website apply to hosting videos as well. Because of this difficulty many organizations overlook video content for their websites.

4.2 A Simple Solution

A simple solution to this problem is to let someone else deal with it. The Internet is full of video hosting websites that let anyone upload video content for free. These videos can then be embedded into the original website, and the web browser will handle the details for loading the video from the secondary site. The visitor will not be able to tell the videos are being hosted by a separate site, but will benefit from a dedicated server and shorter response times.

Google ranks YouTube as the second most visited website on the Internet with over 92 billion page views.[6] YouTube can handle all the video hosting needs for websites small and large. It does not charge to join and makes its money through advertising. Currently, Kansas State University has several channels for various areas of the university. The marching band, football team, and even the radio station have channels; however, the College of Engineering and the Mechanical and Nuclear Engineering Department formerly did not. A channel has been created dedicated to the Mechanical and Nuclear Departments videos. This has provided a central location for all the department videos to be displayed.

4.3 Platform and Encoding Independent Videos

Not all videos are created equally; in fact, there are a large number of variables associated with video creation. This can be a problem when trying to make these videos available to the highest number of people possible. Videos are encoded to reduce the size of the video and make it easier for the video to be transferred. However, encoding a video adds restrictions to how the video can be played. The person trying to watch the video must have the correct software to play this video. Unfortunately, there is no standard method of encoding or playing videos. Several common formats are used, and groups tend to have unofficially adopted a common choice. On top of this, these unofficial standards vary from country to country. What might be a very common format in one area of the world is not frequently used in another.

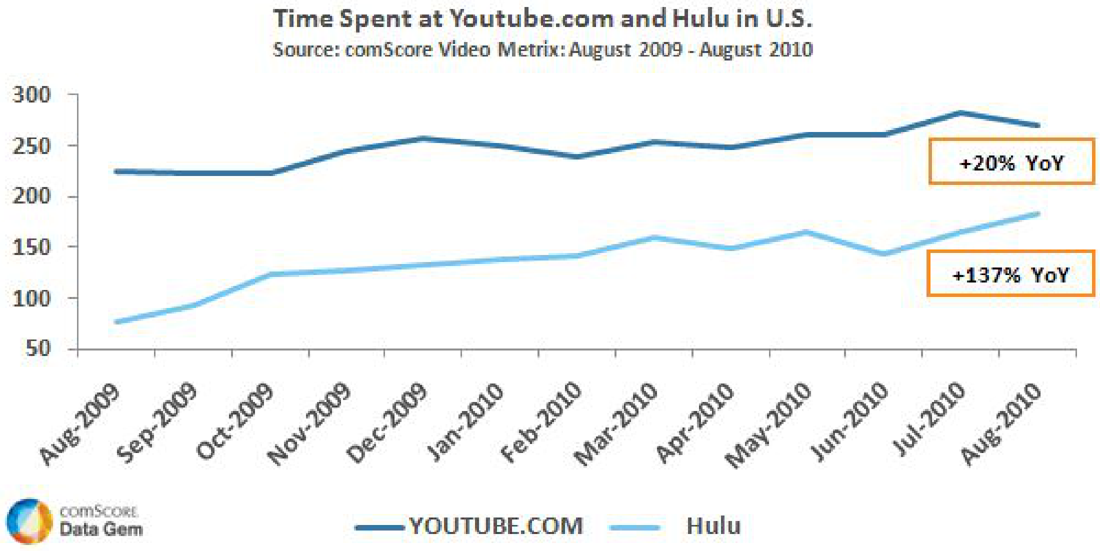
YouTube solves this problem by accepting a large number of video formats. The tools and setting used to create the video are less important. The creator of the video can record the video in a familiar fashion and without the added cost of specialized tools. After the video has been uploaded, YouTube will reformat videos into its standard format, Flash. While anyone can reformat a video on his or her own, this process can be difficult and time consuming. It is best left to dedicated software and people with specialized knowledge of the topic. YouTube has a very efficient method of doing this, and the results are very good. After the video has been reformatted, it can be played by anyone with an Internet browser such as Internet Explorer or FireFox. This make the video accessible to anyone around the world without any compatibility issues.

4.4 Security and Content Monitoring

One issue of hosting videos is the ability to monitor the videos being published. It is desirable to make it quick and easy to make a video public, but how can one monitor that the videos being shown are appropriate? Checking every video for inappropriate speech, behavior or images would be a full-time job. However, screening every video submitted for display would slow the whole process down considerably. YouTube also solves this problem. YouTube screens the videos for copyrighted material, adult content, and various other inappropriate subject matter. It allows any viewer to report a video for these kinds of violations. The video is then reviewed and removed if necessary. While it is unlikely a university department would have a problem with this, it is reassuring to know that this system of checks is in place.

4.5 Increasing Visibility

”Youtube viewers were watching nearly 4.5 hours (269 minutes) of video in August 2010” [10], and YouTube had 490 million unique visitors.[6] Figure 4.1 shows a year-over-year comparison of the time spent watching YouTube and Hulu in the United States. This makes YouTube a huge resource for gaining attention to one’s topic. With every video that someone watches, YouTube makes suggestions based on what other viewers watched. This increases the visibility of the videos to a dedicated audience. Someone watching a similar video may have an interest in this video as well. By helping viewers find videos they want to see, YouTube finds an audience for the video. The description of our videos included a link to the previously discussed website. This will help to draw interested viewers to the new website. This has been particularly useful for the Mechanical and Nuclear Engineering department because it draws prospective research partners, students, and visibility.



Time Spent at Youtube.com and Hulu in U.S.

Source: comScore Video Metrix: August 2009 - August 2010

Figure 4.1: *Minutes spent on YouTube per viewer in the United States from Aug. 2009 to Aug. 2010 and compared year-over-year (YoY)*[10]

4.6 Summary

YouTube offers a simple and free solution to displaying video content on the Internet. Websites can focus on the content they wish to share and not the details of how to share it. YouTube also offers some security that inappropriate material will not be displayed by removing violating videos. Finally, YouTube offers another avenue of drawing visitors to the website by suggesting related videos. These factors make using YouTube a great solution for managing video content on a website.

Chapter 5

Animations

In the controls field we often run simulations of systems. However, the resulting data is usually summed up by a simple graph. This often is not very impressive and can be hard to visualize. Typically, the result is a series of numbers which approach zero or track a trajectory. Simple plots of this data do not provide a good visualization of what the system is doing. However, without much additional work, a three-dimensional animation of the system can be created. The procedure for generating this kind of animation was created, and several animation examples were added to the website. Figure 5.1 is one of these examples. In this section, we will discuss why these animations were created. The specific procedure for creating these animations is detailed in Appendix A.

5.1 Animations Vs. Graphs

Animations have several advantages over graphs when presenting controls data. First, graphs are static images whereas animations change over time. This makes it much easier to understand how the system is responding as time passes. Second, animations can show a three-dimensional representation of the controls system. A graph image is one two-dimensional image and often only focuses on a single part of the system. Finally, animations can show how the entire system responds simultaneously. With graphs, it is necessary to have multiple figures to represent a system. This segments the data and makes it harder to understand how various parts of the system are interacting with each other. Even a simple

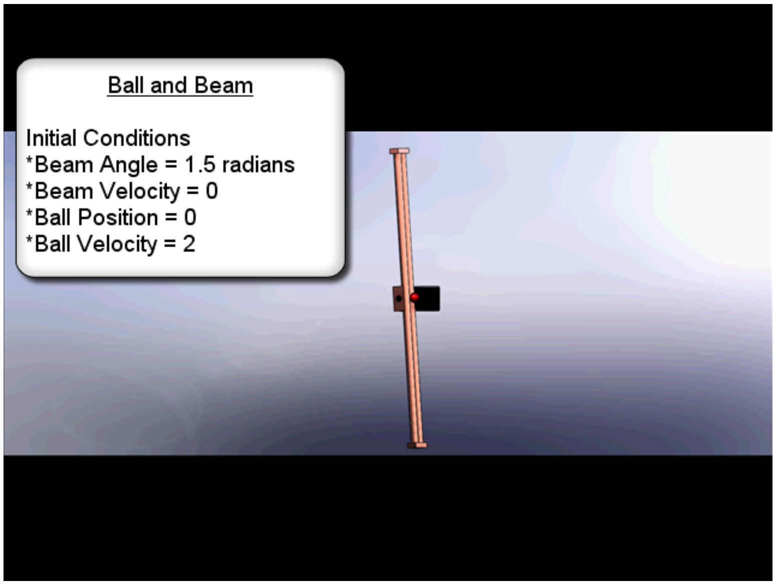


Figure 5.1: *An example of an animation of a simple ball and beam control system*

ball and beam system will have two graphs - one to represent the position of the ball and a second to represent the angle of the beam. Figure 5.2 shows a typical set of graphs for a ball and beam system. To understand how the system is working, these two graphs must be used to create a mental image of how the system is moving. This can be very difficult for most people, and even people who are familiar with the system struggle to visualize the real movement of the system.

5.2 Animation Requirements

Animations do require more work to create than a simple graph. However, the amount of work is minimal in most cases where the preliminary requirements have already been met. Both graphs and animations require data to be generated detailing how the system state evolves. The only additional requirement for an animation is that a three-dimensional model of the system be created. It is not uncommon for this to be done before the simulation is created. Models of the system are used to calculate mass and moments of inertia for more complicated geometry. In addition, if the system is going to be fabricated for real world

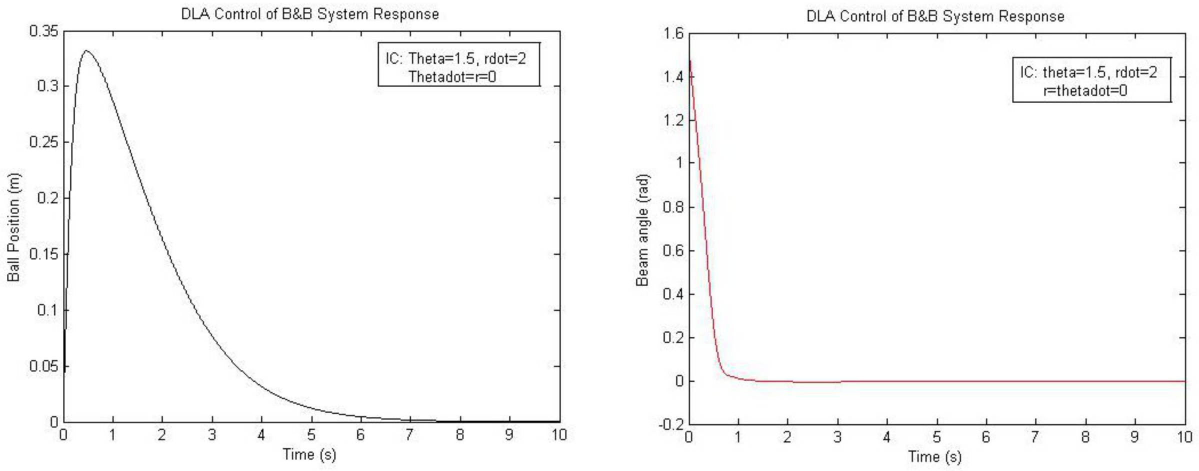


Figure 5.2: *Example graphs of the position and angle of a simple ball and beam control system*

applications, a three-dimensional model is almost always required to aid in the design and construction. Because it is so common to already have a three-dimensional model of the system, it only requires a minimal amount of additional work to create an animation.

5.3 Separation of Encoding Process

Creating the animation can be separated into two parts, creating the animation and encoding it for ease of display. While it is possible to perform these steps simultaneously, it is advantageous to separate them. Each of these processes are time consuming, and even the smallest mistake requires the entire process to be repeated. The first of these steps it to define the animation parameters and record the animation. This is accomplished by defining the movement and then rendering the animation. The rendering will take ten to fifteen minutes for a 30-second animation with moderate quality settings. Longer animations and higher quality settings make renderings that take over an hour very common.

Once the animation is rendered it still needs to be post-processed to encode it into a format easier to share. This process can be performed immediately after the rendering is complete. However, this process is also time consuming and can last for several minutes for short animations. It is very common to complete the rendering and encoding process only to find the animation has a mistake or does not act as anticipated. The direction of motion being backward, the model moves out of screen; or the video does exactly as defined but just doesn't look good. These are all common issues with the animation creation process.

By separating the animation and encoding process, the animation can be viewed and modified without waiting for the encoding to finish. This allows multiple rendered versions of the animation to be created, viewed, and modified before the encoding process ever starts. The drawback to this method is that videos which are not encoded are very large files. Often these are hundreds of megabytes for animations less then one minute in length. However, because the animation process is very computationally intensive, desktop computers with large hard-drives are often used. Once the animation has be finalized, a single encoding can compress the file's size and place it in a format which is easy to share.

5.4 Summary

Animations can be a powerful demonstration tool when teaching about control systems. A three-dimensional representation of the control system can be created, which will show a more realistic representation of the system being demonstrated. These animations are not overly complicated to create and often can be built using already available tools. While the animation process can require several hours of setup and rendering time, often this time can be reduced with proper configuration of the animation. Using the correct amount of data and separating the rendering and encoding process make creating animations fairly straightforward. Based on this information the benefits of creating animations far outweigh the cost.

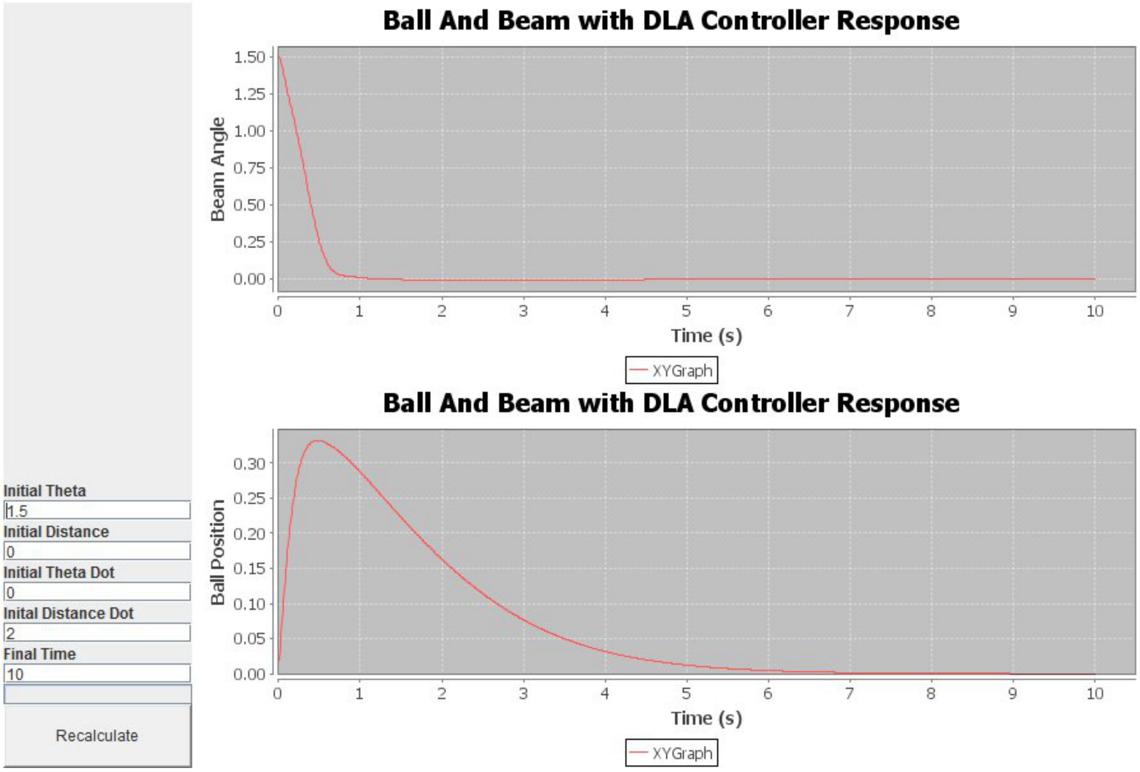
Chapter 6

Interactive Graphs Through Applets

In addition to a website with Web 2.0 features, interactive graphs were created using Java applets. As a default, these graphs show data from control simulations which had been created previously in MatLab. However, rather than simply always showing a single set of data, the applets are able to perform the simulation and graph the resulting data given viewer specified initial conditions. A procedure was created to generate these interactive graphs and is included as Appendix B.

6.1 Interaction

Methods which stimulate the widest variety of senses will generally be the most effective.[9] Graphs are the most common method of displaying numeric results; however, they are static and do not allow the viewer to interact. Interactive graphs can give the viewer more control of the data, such as resizing the graph and zooming in on specific areas of the data for added detail. Additionally, the graphs of controls systems allow the users to change the initial conditions and generate new graphs. Because viewers are interacting with the graph, their understanding and retention of knowledge will be increased.



Initial Theta
1.5
Initial Distance
0
Initial Theta Dot
0
Inital Distance Dot
2
Final Time
10

Figure 6.1: *An example interactive graph of the position and angle of a simple ball and beam control system*

6.2 Variable Initial Conditions

Interactive graphs can be created and then displayed on a website. Viewers are able to recalculate the simulation and the graph is plotted interactively. Additionally, the users can change the initial conditions of the system and a new graph can then be generated. This allows the users access to an infinite number of graphs, instead of just one. It would be impossible to generate every single set of data for a control system, and even generating a wide variety of graphs would create difficultly in how to present this data to the user. However, with an interactive graph the users can generate any set of data, within the scope of programmed variables, they desire.

By allowing users the ability to change the initial conditions of a particular system, they gain a much more in-depth understanding of the control system. Rather then seeing a single snapshot of how the system acts, they can experiment with the system, seeing where the system becomes unstable. Experiential learning involves a ”direct encounter with the phenomena being studied rather than merely thinking about the encounter, or only considering the possibility of doing something about it.”[2] While still not as good as working with a real system, interactive applets bring the user one step closer to understanding the control system.

6.3 Applet Graphs Require More Work

While applet graphs offer the user more information about a system, they require more work to create than a simple graph. Java applets require programming the control system and graphing functions. They also require the creation of an HTML page to host the applet. While the host pages can be very simple, creating the applet itself requires quite a bit of work. Additionally, Java programming is not a skill which most controls engineers or students process.

The task can be made much simpler with two open-source libraries which add needed functionality to Java. The first is the JFreeChart Library which creates graphs.[5] This library makes it possible to create very impressive and functional graphs with little knowledge of Java programming. The second open-source library is the Apache Commons Mathematics Library.[4] This adds much needed mathematical functions to Java; specifically, the Runge Kutta integrator of order 4 and 5. This is the function used to solve the ordinary differential equations for most control systems. Both of these libraries can easily be downloaded from their respective websites.

With these tools, it is a much less daunting task to program an interactive graph for a given controls system. While still requiring more work than generating a static graph of the data, the additional demonstration and educational value of an interactive graph is worth the added work. Appendix B details the procedure for creating interactive graphs using the above Java Libraries.

6.4 Summary

Interactive applets are a method to increase the interaction between the viewer and the information being presented. Because it is often impossible to interact with a real world control system, a suitable representation must be made. While traditional graphs have been the preferred method for displaying the results of a controls system, interactive applets can bring the viewer one step closer to the real world simulation. Much more information can be made available through interactive applets, because they can generate an infinite number of simulations. This interaction and wider range of data will increase the understanding for the viewer.

Chapter 7

Conclusion

The main area of work for this thesis was the development of a new website using Web 2.0 tools. This new website is easier and quicker to update, allowing new content to be added as soon as it becomes available. User accounts can be created, allowing multiple people to update the site at the same time. In this way, anyone given permission to update the site can add his or her area of research, detailing the work and displaying content such as graphs, videos, and even simulation files.

Because the new website is collaborative, it has the potential of becoming a single location to learn about all the controls research happening at Kansas State University. In this way professors will have access to use this information as teaching aids for their classes. Additionally, students will be able to use the website as an independent learning tool, learning more about topics they are interested in. Graduate students looking for research opportunities can use the website to find possible topics for their graduate work. Kansas State University is not the only place controls research is happening. Many universities and groups need a place to display the work they are doing. By making the website collaborative, it is possible to have research from all around the world in one location, while still maintaining control over the content.

Videos are a common method of demonstrating a working control system. These videos can be very impressive and are one of the best ways to generate interest in a topic from both students and potential funding organizations. However, it is has been difficult to make these videos available. Supplying these videos on a website has presented problems such as slow loading and specialized video playing software. Current social video-hosting websites like YouTube allow viewers to play these videos from any web browsing software. By taking advantage of this resource, these problems were solved and the visibility of these videos increased.

Control systems were difficult to demonstrate on the older websites. It was also impractical to bring some control systems to the classroom. A method of creating three-dimensional animations of controls systems was developed. These animations were created using models and simulation data which were already available. By showing the system in a three-dimensional animation, viewers can understand how these control systems behave.

A method for creating interactive graphs was developed. Traditional graphs present a small set of data about a control system. The graphs are static, making it difficult to understand how a system reacts to a wide variety of situations. The new interactive graphs allow the user to adjust the initial conditions of the system and simulate them directly from the website. By increasing the interaction with the viewer, memory retentions and comprehension will be increased.

To summarize, the methods and tools discussed previously offer a wide range of features to the users. By making the website easier to update, more content can be added. By creating a single place for this content, it will become easier to find and make use of the content. Additionally, by creating higher quality content such as streaming videos, animations and interactive graphs, the website can better demonstrate the quality of research being done at Kansas State University and other universities.

Bibliography

[1] Cliff B. Web host's research so small firms update websites infrequently. *Retrieved from http://www.tophosts.com*, August 2010.

[2] L. Borzak. *Field Study. A source book for experiential learning.* Beverley Hills: Sage Publications, 1981.

[3] Adrian Bowyer. Reprap. *Retrieved from http://www.reprap.org*, 2010.

[4] Apache Commons. Commons-math: The apache commons mathematics library. *Retrieved from http://commons.apache.org/math/*, 2011.

[5] Thomas Morgner David Gilbert. Jfreechart. *Retrieved from http://www.jfree.org/jfreechart/*, 2011.

[6] Inc Google. The 1000 most-visited sites on the web. *http://www.google.com/adplanner/static/top1000/*, December.

[7] TikiWiki CMS Groupware. Tikiwiki. *http://www.info.tiki.org/tiki-index.php*, 2011.

[8] Jamie Lyons. What are the top ten most visited sites of 2010? *http://www.sitepronews.com*, December.

[9] Vernon A. Magnesen. A review of findings from learning and memory retention studies. *Innovation Abstracts*, V5 N25, September 1983.

[10] ComSource Video Metrix. Time spent on youtube.com and hulu in the u.s. *http://www.comscoredatamine.com/2010/09/minutes-per-viewer-at-youtube-com-and-hulu-in-u-s/*, September.

[11] Prashant Sharma. Core characteristics of web 2.0 services. *http://www.techpluto.com*, November 2008.

Appendix A

CREATING VISUALLY IMPRESSIVE ANIMATIONS FOR CONTROLS SYSTEMS

The following appendix was written as a stand alone document and has been included for reference. The original formating has been preserved.

CREATING VISUALLY IMPRESSIVE ANIMATIONS FOR CONTROLS SYSTEMS

COMBINING MATLAB DATA AND SOLIDWORKS 3D MODELS INTO AN ANIMATION

Joshua C. Updyke
Kansas State University
Mechanical Engineering Department
jupdyke@ksu.edu

Warren N. White
Kansas State University
Mechanical Engineering Department
wnw@ksu.edu

ABSTRACT

In the controls field we often run simulations of systems. However, the resulting data is usually summed up by a simple graph. This is often not very impressive and can be hard to visualize. Often the result is a series of numbers which approach zero. Simple plots of this data do not give a good visualization of how the system will respond. This paper will guide you through the steps needed to create visually impressive 3-dimensional animations of a controls system.

It is important to point out that while this paper focuses on using SolidWorks and Matlab to create the animation, other software packages can be used. In addition, while the focus of this paper is on how to create an animation of a controls system, the same practices can be used to make any sort of data-driven animation.

1. INTRODUCTION

Before discussing the details of each aspect of creating an animation, the overview of the process will be covered. Before creating the animation two things are needed. A set of data which details how the system will act, and a three-dimensional model of the system. This paper will not go into detail about how to create either of these. There are many papers, tutorials, and books about how to create them. At the end of the paper some information which may help is included. The animation will combine the simulation data and the 3 dimensional model. By doing this, an animation that behaves the same as the simulation but has the impressive graphics of the three-dimensional model can be created. Below are the main steps in creating the animation:

- Create the three-dimensional model
- Generate the data
- Formate and import the data
- Set up the animation
- Rough render the animation
- Final render the animation
- Post process the animation (optional)

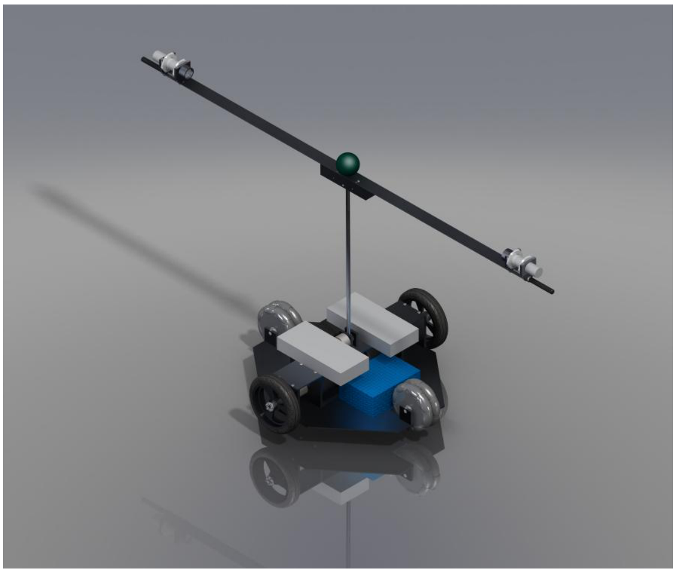


Figure 1: Animated Control System

2. PREPARING THE DATA

The data is detailed for the SolidWorks 2010 format. Other CAD packages may require the data to be formatted slightly differently. However, the fundamentals should all be the same.

2.1 What data is needed?

When creating simulations, it is common for us to look at various sets of data. Time, position, velocities and accelerations are all common. However, when creating animations, only time and position are needed. It is possible to use velocity instead; however, it is simpler to use position and that is what this paper will focus on. Positional data for each moving part of the system will need to be generated and each positional value must have a time value.

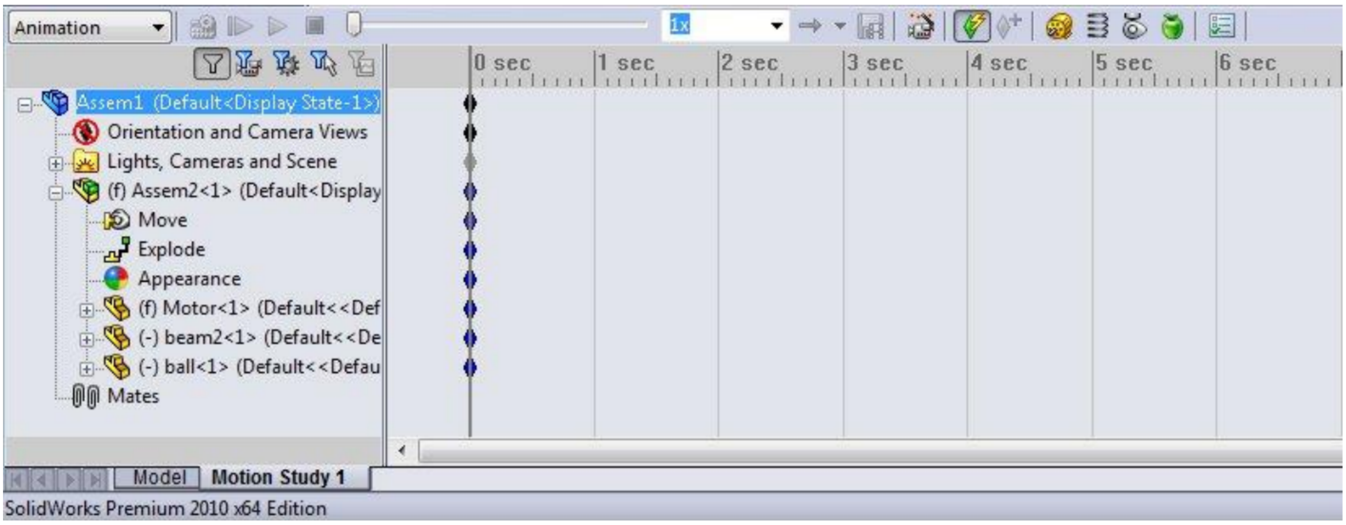

Animation
1x
0 sec
1 sec
2 sec
3 sec
4 sec
5 sec
6 sec
Assem1 (Default<Display State-1>)
Orientation and Camera Views
Lights, Cameras and Scene
(f) Assem2<1> (Default<Display
Move
Explode
Appearance
(f) Motor<1> (Default<<Def
(-) beam2<1> (Default<<De
(-) ball<1> (Default<<Defau
Mates
Model
Motion Study 1
SolidWorks Premium 2010 x64 Edition

Figure 2: Motion Study Button Location

The time vector does not need to have a constant time-step. It is common to use a variable step runge-kutta method when running simulations. SolidWorks will interpolate the data between each data point.

2.2 How much data?

Typically, more data is better. However, when making animations, that is not actually true. In fact, too much data can lead to long set-up and rendering times, without creating a noticeable difference to the animation. The basic idea is that only about 30 points of data for each second of animation is needed. It is common when creating an accurate simulation that a small-time step, perhaps 0.01 seconds or smaller, is required. If this is the case, the data can be sampled to reduce it.

Example: If a constant time-set of 0.01 seconds is used and a simulation for 20 seconds is created, 2000 points of data will be generated. However, when creating the animation, a frame-rate of 30 frames per second is used. This means only 600 frames will be animated, and only 600 of those 2000 data points is really used. The rest will just slow down the process of making the animation, because the software has to decide which data points to use.

2.3 Coordinate system, units, and precision

SolidWorks uses absolute positioning when creating animations. The locations of all the parts of the model when the animation is started is considered to be zero. It is common for simulations to start the system some place other than the origin. If the data defines the origin at different location than the starting position, the data will need to be adjusted.This can usually be done by subtracting or adding the starting location to all the data points.

Because most data is simply two columns (time and position), it is important to understand the direction of motion. Since the data does not inherently contain any information about direction, it must be defined later in SolidWorks. It may be helpful to create a diagram which defines the origin, coordinate systems and which direction is positive for each data set. This will be helpful for creating the simulation as well as the animation. The data also does not contain any units. It is important to ensure the units used when generating your data are the same as the units used when importing the data into SolidWorks. When importing data into SolidWorks, the data cannot have more than one data point for each time. This is a problem that can arise unintentionally from a lack of precision. When exporting data from MatLab and attempting to directly import it into SolidWorks, will cause an error referring to duplicate data. This is caused by only the first 3 decimals of the data being read. Opening the data file in Microsoft Excel or OpenOfﬁňĄce Calc and changing the format of the cells to show more decimals solves the problem.

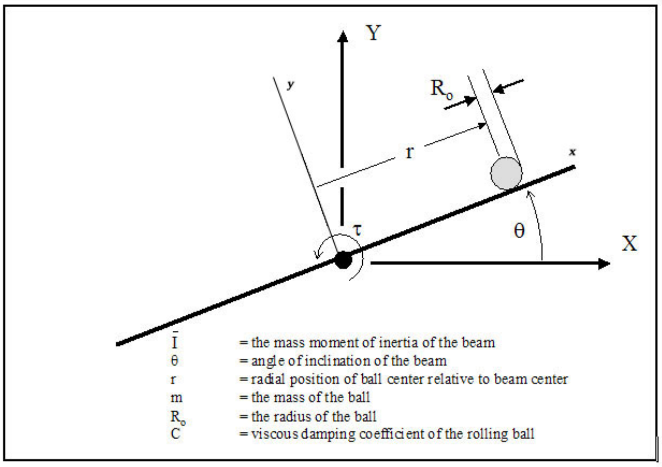
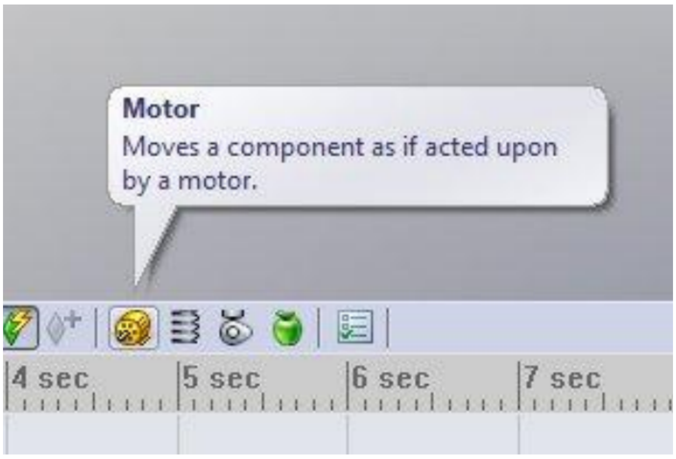

Y
y
R$_o$
r
x
τ
θ
X
Ī = the mass moment of inertia of the beam
θ = angle of inclination of the beam
r = radial position of ball center relative to beam center
m = the mass of the ball
R$_o$ = the radius of the ball
C = viscous damping coefficient of the rolling ball

Figure 3: Example Coordinate System Diagram

3. PREPARING THE MODEL

Many times 3d models have already been created before the simulation is run. However, this model may not be best suited for creating animations. The model does not need to be complete or 100 percent accurate. In fact, it is often intentionally incomplete and only as accurate as needed.



Motor
Moves a component as if acted upon by a motor.

Figure 4: Example Coordinate System Diagram

Sometimes it is hard for engineers to think like animators. The simulation has already determined how the system will act. The 3d model only needs to show how the system looks.

3.1 Performance issues

It is possible to improve the performance by making some changes to the model. A model used to create an animation and a model used to for constructing a system have different requirements and as such can be very different. Most models are created to be as accurate as they can be. They may contain each and every part of the system down to the nuts and bolts. They may have parts that are extremely detailed and contain hundreds of features. While these aspects make a very accurate model, it may not create the best animation. It is possible to use this model however.

A better model for this application would be one that kept all the aspects needed for the animation but removed those that were not needed. This can greatly increase the speed of rendering the animation. Because the model must be updated each frame of the animation, having fewer parts to translate each time will make the animation process much faster. Here are some guidelines for a good animation model:

- As few parts as needed, leave out any internal parts that cannot be seen.
- Use cosmetic threads instead of real threads.
- Parts with less complex features are better than those with complex features.

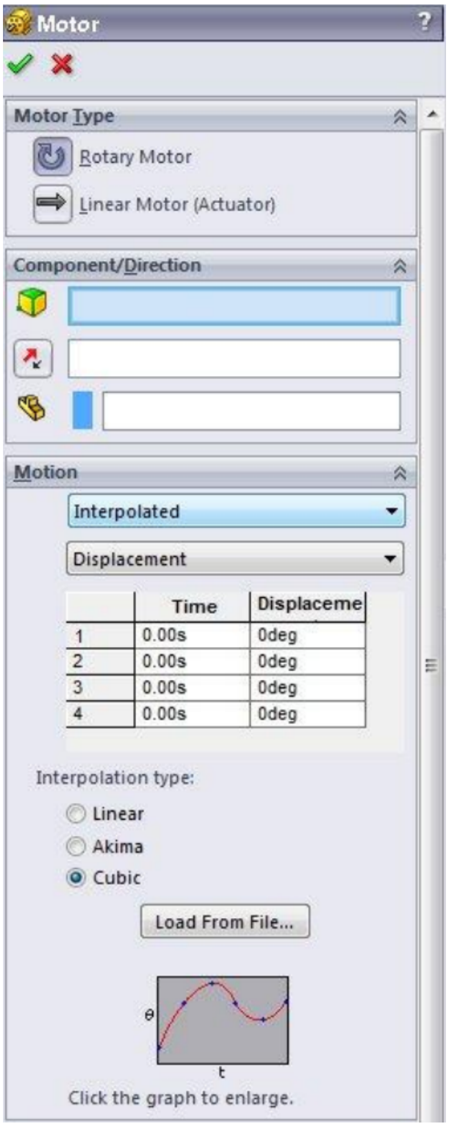
Make sure the only parts that can move in the model are those being defined with data. Everything else should be fully constrained.

4. CREATING THE ANIMATION

The first step in creating the animation is to click the motion study button at the bottom left of the SolidWorks screen (Figure 2).

4.1 Motor definition

Next, a motor is added for each moving part. This will define how that part moves. Click the motor button shown in Figure 3. The motor definition screen is where the movement of each component of the animation is defined. First, define the motor as either a rotary or linear motor. Next, select the component which will move. For rotary motors select an axis about which the component will rotate about. For linear motors selecting an edge that is parallel to the direction of motion works well.



Motor

Motor Type
Rotary Motor
Linear Motor (Actuator)

Component/Direction

Motion
Interpolated
Displacement

	Time	Displaceme
1	0.00s	0deg
2	0.00s	0deg
3	0.00s	0deg
4	0.00s	0deg

Interpolation type:
○ Linear
○ Akima
● Cubic

Load From File...

Click the graph to enlarge.

Figure 5: Motor Definition

This step can be a little difficult, and generally trial and error is the best solution. After a component is selected, an arrow will be displayed on the screen to show you how the component will move. Note that the second button in Component / Direction box allows the direction of motion to be reversed. Your data does not contain direction, so you must choose the direction of motion that matches your simulation. Finally, if the direction of motion is relative to another component, use the final box to select this component. This is often the most complicated part of the animation.

Selecting "Interpolated" as the motor type will allow you to define the movement of the motor by an external file. It is best to select ”Displacement” as the data type. However, velocities and accelerations can also be used. It is important to remember at this step that SolidWorks will assume the data is in the default unit for measurement of the type defined. For displacement, SolidWorks will assume the data is the default unit of measurement for distance (inches, centimeters, etc.) To change it, change the units of the assembly in the SolidWorks options page.

Next, choose the interpolation type. For small time-steps there will be little difference between them. Finally, click the ”Load From File” button, and locate the data file. This may take a while to load into SolidWorks, especially if the file contains a large amount of data. Very large data files will load very slowly and can even cause SolidWorks to crash.

Once the data is loaded, click the green check mark to finalize your motor definition. The animation will now have a motor added to the time-line. A motor will need to be defined for each moving part of the animation. It can be helpful to simulate the animation after each motor definition to make sure it acts as expected. It is highly recommended to save your assembly after each motor definition.

5. RENDERING THE ANIMATION

This section discusses issues related to transforming rendered model images into usable fluid video. Through careful planning several pitfalls can be avoided.

5.1 File formats

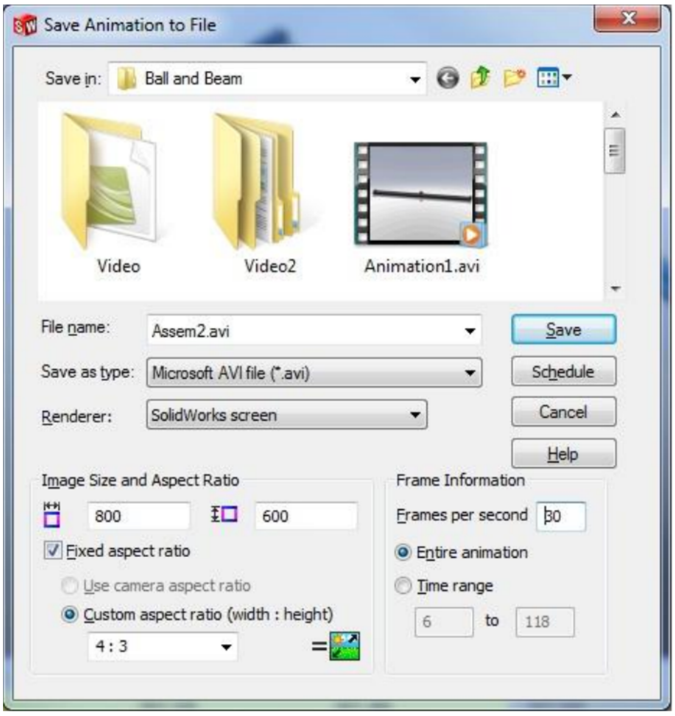
Animations can be in many different file formats. It can be a movie or a series of image files. Generally a movie file is desired. The most common file format for this is Audio Video Interleave (.avi). Other options would be RealMedia (.rm), MPEG (.mpg, .mpeg, .mp1, etc) or Flash animation(.swf). Each of these has its own benefits. However, unless there is a specific reason, it is recommended that the .avi file format is used. SolidWorks 2010 can create these files easily, and the file format can always be changed later using a post processor.

5.2 Video size

The dimensions of the video will greatly impact the final file size of the animation and the quality of the video. If the video dimensions are too small it will be hard to see; Too large a file will be cumbersome to download. SolidWorks allows the image size and aspect ratio of your final video to be defined. A 4:3 ratio is a standard video and print format and a good place to start. Additionally, dimensions of 800 by 600 pixels create a modest file size. It is large enough to see details of the animation while keeping the file size manageable.

5.3 Frame rate

One of the most important settings when creating an animation file is the Frames Per Second (FPS) setting. The default FPS is 7.5 which can leave the video a bit choppy. The higher the FPS, the smoother the motion in the animation. However, the rendering time will be longer and the file will be larger. Good results have been found using 20-30 FPS. Because rendering the animation can take hours, it is a good idea to render the animation at a lower frame rate first to make sure it looks correct. It is common to create several ”rough” renderings before the final rendering. This can seem tedious, but rendering a video for hours only to find that part of the animation travels off the screen will waste more time in the long run.



Save Animation to File

Save in: Ball and Beam

File name: Assem2.avi

Save as type: Microsoft AVI file (*.avi)

Renderer: SolidWorks screen

Image Size and Aspect Ratio

800 600

☑ Fixed aspect ratio

○ Use camera aspect ratio

● Custom aspect ratio (width : height)

4 : 3

Frame Information

Frames per second 30

● Entire animation

○ Time range

6 to 118

Figure 6: Animation Settings

5.4 Optional photo-realistic rendering

SolidWorks includes the option to photo-realistic render your model. This is normally used to create a single photo quality image of a model. However, it is possible to render each frame of an animation this way. The result is a more realistic looking animation. Rendering each frame of the animation this way will take much longer than using the default SolidWorks screen option. Fortunately, setting up a rendering and animation task to automatically run overnight using the SolidWorks Task Scheduler is possible. If this method is used, rendering the animation with the SolidWorks screen first to make sure the results are as expected is recommended. Also use PhotoWorks to render several different static images of the assembly before beginning. A typical photo-realistic rendering will take about ten times longer then the SolidWorks screen.

5.5 Video Compression

Video compression is a trade-off between disk space and video quality. A good animation will be small enough that it is easy to transfer. It may be desirable to e-mail the animation or host it on a website without difficulty. If the file is too large, than it will take a long time to download which could deter the audience from watching it. SolidWorks offers several different encoding methods when creating an animation:

- Microsoft Video 1
- Intel IYUV Codec
- Logitech Video

- Full Frame (Uncompressed)

Rendering the animation with Full Frame (Uncompressed) is suggested if post processing is available. Otherwise, Microsoft Video 1 is suggested. SolidWorks does a decent job of encoding the animation; however, it is not without problems. It makes sense to segment the animation process at any point possible. If the animation is rendered uncompressed, the animation process and encoding process are separated. This way a problem with the encoding does not require that the animation be recreated. This is particularly helpful when creating longer animations with photo-realistic renderings, where the animation process is quite long. Uncompressed animations create very large files, and that should be considered before beginning.

6. POST PROCESSING THE ANIMATION

If the animation is created with Full Frames (Uncompressed) option, it will need to be post processed to encode it into a format that is better suited for distribution. There are a large number of free programs that can do this. Alternatively, very good success has been had using a relatively inexpensive program called Camtasia by Techsmith. Because it is designed for video editing, it has many more options than SolidWorks. It can even encode your animation into Flash and create an .html ïňĄle to play the video directly from your website without the need of downloading.

Camtasia can combine several video files into a single file as well. This is helpful when trying to create very complicated animations, because everything does not need to be defined in a single animation in SolidWorks. Making multiple short animations and splicing them together once they are video files requires less computer resources than defining everything at once in SolidWorks. It also reduces the chance of a small error ruining hours of rendering time.

6.1 Finishing Touches

A few final options are available during the post processing stage to make the animation more impressive. Music or sound effects can be added to the animation. A simple background music will have a dramatic effect on the animation. Title screens and call-outs can also be added. Jumping directly into and animation is disorienting and a title screen which introduces what the video is and who created it will give the audience the needed preparation to watch the video. Call-outs are a nice way to explain what is happening in the animation, and most video editing software can add them.

7. CONCLUSION

By following these steps stunning animations can be made easily. These animations will give a much more intuitive and impressive representation then a simple graph.

APPENDIX

A. DATA REDUCING MATLAB CODE

This code should be used when the data contains too many data points to be useful for creating an animation. It is common to have around 30 frames per second of video. This means only 30 data points for each second are needed. It is not uncommon to use more data than this, but the computational speed of the computer needs to be considered. Excel also has limits on the number of data points which can be inputed. Because of these reasons, it is recommend to keep the data less then 300 data points per second of video. The example data for this paper had many more data points than needed. This section of MatLab code was used to sample the data in order to reduce it.

```
scale = 50;
old_data = data;
[r,c]=size(old_data)
count=1;
while (count*scale) < r
new_data(count,:)=old_data(count*scale,:);
count=count+1;
end
Reduced_data = new_data;
[r,c]=size(new_data)
```

B. ADDITIONAL RESOURCES

Here are some additional resources that might be helpful when creating animations.

B.1 SolidWorks Modeling

- SolidProfessor is a SolidWorks Solution Partner delivering the largest library of SolidWorks video training and tutorial courses on the market.
- http://www.solidprofessor.com/
- Carnegie Mellon SolidWorks Lessons
- http://www.me.cmu.edu/academics/courses/NSF_Edu_Proj/Statics_Solidworks/

B.2 Simulation

- Kansas State University's Non-Linear Controls Website
- http://www.mne.ksu.edu/static/nlc/
- Carnegie Mellon's Control Tutorials for MatLab
- http://www.engin.umich.edu/group/ctm/

B.3 Post Processing and Encoding

- Camtasia Studio screen recorder and editor for Windows
- http://www.techsmith.com/
- Any Video Converter Freeware
- http://www.any-video-converter.com/

Appendix B

CREATING JAVA APPLETS FOR CONTROLS SYSTEMS

The following appendix was written as a stand alone document and has been included for reference. The original formating has been preserved.

CREATING JAVA APPLETS FOR CONTROL SYSTEMS

TURNING MATLAB SIMULATIONS INTO INTERACTIVE WEBSITES

Joshua C. Updyke
Kansas State University
Mechanical Engineering Department
jupdyke@ksu.edu

Warren N. White
Kansas State University
Mechanical Engineering Department
wnw@ksu.edu

ABSTRACT

In the controls field we often run simulations of systems. However, the resulting data is usually summed up by a simple graph. This is often not very impressive and can be hard to visualize. By using the Java programming language an interactive graph can be created. This graph is capable of not only displaying a single set of data, but generate and graph thousands of sets. The user can adjust the initial conditions of the control system and graph the resulting data. This allows the user access to much more information and by interacting is more likely to remember the concepts demonstrated. This paper will explain the advantage of interactive graphs and detail the tools used to create them. Figure 1 shows an example of one of these interactive graphs.

1. INTRODUCTION

Simulating controls systems is an integral part of understanding new systems, teaching controls and demonstrating the effectiveness of the system. MatLab is the common tool for creating these simulations because it can create them effectively and quickly. However, it can be difficult to share a simulation with others because MatLab is a scripting language. This means the recipient must also have MatLab, and the differences in version may cause code run incorrectly. This project had two goals. First, to make distribution of the simulation as easy as possible. Second, to allow the audience to interact with the simulation. Most simulations created in MatLab simulate the system with a single set of initial conditions. This does not show the true potential of a good controller. A good controller must be able to control the system over a wide range of situations and initial conditions. While most MatLab simulations are capable of simulating over many initial conditions, the user must know how to change those conditions and be familiar with Matlab to do so. An easily distributed and interactive method of displaying the control system was needed.

2. WHY JAVA?

Java was choose to achieve both of these goals. Java is a general purpose object oriented programming language. It can run on any Java Virtual Machine (JVM) regardless of computer hardware or software and Java is widely used in web applications. Almost all web-browser software is capable of running Java. This means that anyone who can browse the Internet could interact with our simulation without downloading or installing any software.

Java applets are small programs that run inside of a web-browser and provide interactive features to web applications that can not be provided by HTML alone. An applet can have buttons, check boxes and even capture mouse movement. These features allow the creation an interactive simulation for the user. As apposed to a static simulation using MatLab code, or worse a image of the results which provide no option of interaction.

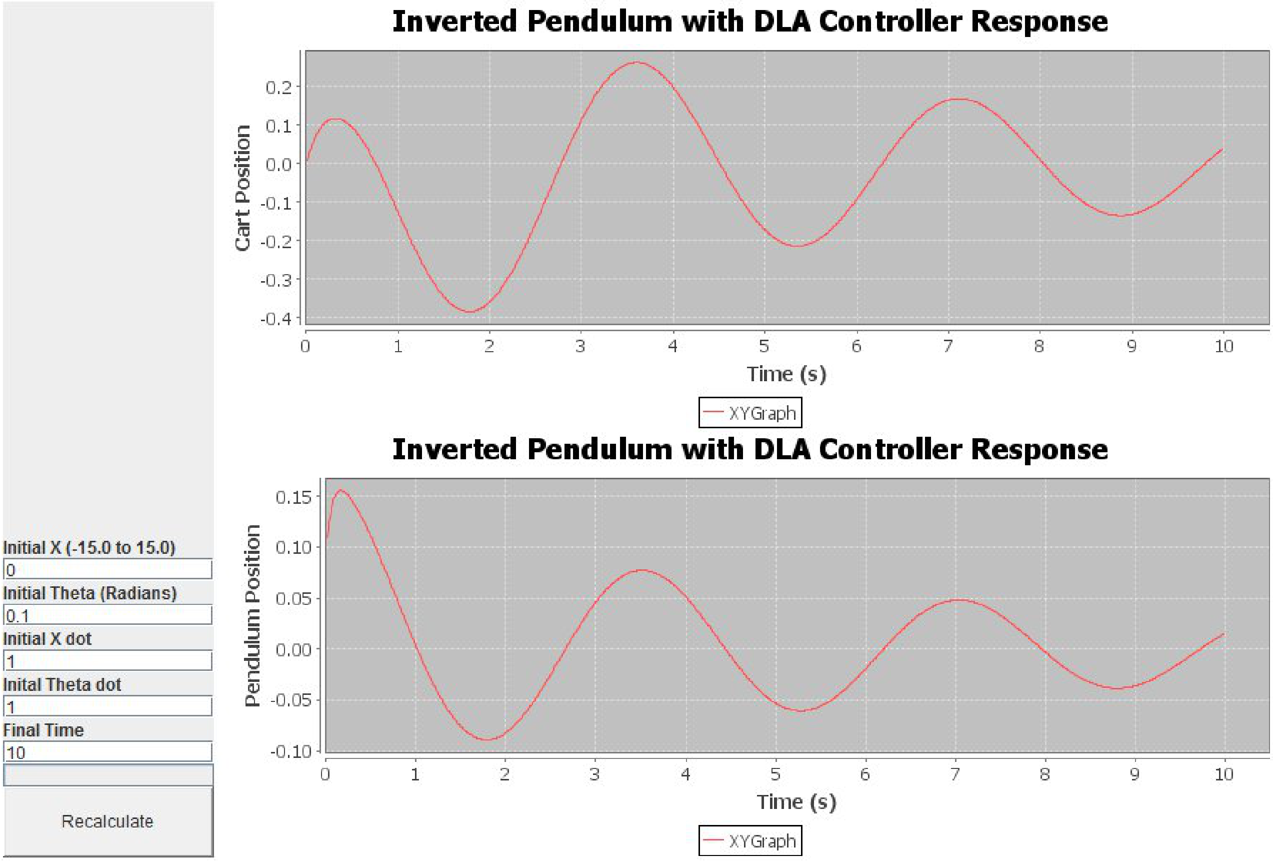
3. NETBEANS IDE

In order to create an Java applet an integrated development environment(IDE) is needed to write the program, compile and test it. NetBeans, Eclipse, JBuilder and Sun Java Studio are all popular choices amount the free and low-cost options. While any Java development environment will work, NetBeans IDE 6.9.1 is used for the simulations created in this paper. NetBeans is a very powerful and easy to use development environment and excellent documentation is provided, which is very helpful for those new to Java programming. The IDE and documentation can be found at `http://netbeans.org/index.html`.

NetBeans can be installed with various supported technologies and can be used to program a long list of languages, including C/C++, Java, and Ruby. NetBeans has several versions tailored to different programming tasks, as well as an all inclusive version. This makes NetBeans an ideal environment for engineers because it supports many projects. By becoming familiar with a single development environment engineers can avoid setting up and becoming acquainted with a new IDE for each language. However, since the goal of this project was a basic Java applet, only the Java SE version is needed.

4. JFREECHART LIBRARY

MabLab has some very powerful graphing tools, which make it very easy to create impressive graphs of simulation data without writing much code. Java does not contain these tools, and programming ourselves them would be time consuming. JFreeChart is a free Java chart library that makes it easy for developers to display impressive charts



Initial X (-15.0 to 15.0)
0

Initial Theta (Radians)
0.1

Initial X dot
1

Inital Theta dot
1

Final Time
10

Figure 1: An example interactive java applet.

in their applications. With JFreeChart it is easy to create charts in Java in almost the same fashion MatLab creates charts. Pie, bar, scatter, histogram, and line charts can all be made very quickly. By making the graphing functions as simple as possible, and similar to a familiar simulation tools like MatLab, the learning curve for creating applet simulations is reduced.

In addition to making it easy to create plots, JFreeChart allows the charts to be edited by the viewer. By right clicking on any chart created with JFreeChart several options display options become available. Series of data can be selectively displayed, colors can be changed, legends can be added or removed, as well as many other display settings can be adjusted to show the data in the most effective manner. Sections of the chart can be zoom in on and the range of the axes can be changed. The charts can even be saved as an image file. These options make JFreeChart much more useful to the viewer then a basic chart. With a very small amount of programming a large number of graphs can be generated and user can decide how best to display the data.

The JFreeChart library is available to download for free from `http://www.jfree.org/jfreechart/`. A JFreeChart Developers Guide is also available in PDF format for $40 from the JFreeChart author. This price is certainly reasonable for supporting an open source project. Other tutorials are easily available and one is listed at the end of this paper. This paper will only cover the very basics of using this library and some example source code is provided at the end. However, the goal of this paper is not to explicitly explain the ins and outs of programming with this tool, but rather give an overview of its functions and how they can be used to demonstrate a controls system.

JCommon is a java class used by JFreeChart, Pentaho Reporting and several other projects. It contains various classes that support user interface, custom layout managers, serialization utilities and many other things. JCommon and JFreeChart are both licensed under the terms of the GNU Lesser General Public License (LGPL) version 2.1 or later. For the purpose of this paper it is only important to know that JFreeChart makes use of this class library and we must include it for JFreeChart to work. This class library is available for downloading from `http://www.jfree.org/jfreechart/`, as well as several other places.

4.1 Programming with JFreeChart

Before you can start programming these two libraries need to be added to NetBeans. At the time this paper was written jcommon-1.0.16.jar and jfreechart-1.0.13.jar were the current versions. However, the version numbers should be replaced with the versions being used. Once the libraries have been downloaded they can be added using the Library Manager, located on the Tools menu of NetBeans. This tool makes adding libraries easy and NetBeans will use the library information to provide documentation support (JavaDoc). In addition, NetBeans will ensure the library is available on the class-path when executing the application.

Other development environments will have similar methods for adding libraries and should be relativity simple. Once the libraries have been added to NetBeans they can be used to write a simple graphing applet and run on the local machine. However, the jar files must be included when distributing the applet and this will be discussed in further detail later.

The following is an example of how JFreeChart can be used to create a pie chart. Only the code related to using JFreeChart is included below. The entire source code is included at the end of this paper and is titled PieChartExample.java. In this example a dataset is created using the DefaultPieDataset() class, and data is added using the setValue() method. Each type of chart in JFreeChart has one or more corresponding dataset classes used to define the data the chart will plot. These are very straightforward and easy to use.

After the dataset is defined and populated the chart object is created using the ChartFactory class. This example creates a 3D pie chart using the createPieChart3D() method. The ChartFactory class contains methods to create a wide variety of chart styles. After the chart has been created it is plotted using any of the matching plotting classes. These classes are used to change the way the chart is displayed. These basic steps are all that is needed to format and display information using JFreeChart. Figure 2 shows the chart created by the following code.

```
Private  PieDataset createDataset() {

DefaultPieDataset result = new DefaultPieDataset();
result.setValue("Engineering", 28);
result.setValue("Computer Science", 21);
result.setValue("Physical Sciences", 20);
result.setValue("Mathematical Sciences", 12);
result.setValue("Life Sciences", 6);
result.setValue("Business/Social Sciences", 6);
result.setValue("Other", 7);
return result;
}

/* Creates a chart
* This is where the chart object is created
*/
private JFreeChart createChart(
           PieDataset dataset, String title) {

JFreeChart chart = ChartFactory.createPieChart3D(
title,   // chart title
dataset,                // data
false,                   // include legend
true,
false
);

//These define aspects of how the chart will look
PiePlot3D plot = (PiePlot3D) chart.getPlot();
plot.setStartAngle(290);
plot.setDirection(Rotation.CLOCKWISE);
plot.setForegroundAlpha(1f);
return chart;
}
}
```

5. APACHE COMMONS MATHEMATICS LIBRARY

The Apache Commons Math library is a lightweight, self-contained mathematics and statistics library addressing the most common math tools not natively available in the Java programming language. This library adds the mathematical functions which are used to simulate a controls system. The most important of these functions is the Runge-Kutta integrator of order 4 and 5. This is the function used to solve ordinary differential equations for the control systems. Additionally, this library adds support for matrices. However, because the systems used for this paper contained only a few inputs and outputs the matrix functions were not needed. However, for larger systems these functions would greatly reduce the length of the code needed to simulate the system.

The following section of code demonstrates how the Runge-Kutta integrator can be used. The entire code is included at the end of this paper and titled ApacheMathExample.java. First, objects are created for both the class defining the ordinary differential equation and the integrator. Then a set of variables are created to hold the start time, current time, y and derivative of y. One key difference between the Apache integrator and MatLab's ODE45 is that you can not pass a time vector to the function. In this way the DormandPrince45Integrator will only return the value of the equation at the final time. To collect data in the intermediate steps we must add a loop, and call the integrator multiple times. Each pass of the loop will solve the differential equation for a small time step, 0.01 in this example. At the end of each pass the example prints the result to the screen. However, when JFreeChart is used this data can be added to the dataset and easily graphed.

```
OrdinaryDifferentialEquations ODE = new OrdinaryDifferentia
DormandPrince54Integrator integrator = new DormandPrince54I

int N = 20;
double t0=0.0;
double t;
double[] y = new double[1];
y[0]=2;
double[] dy = new double[1];
dy[0]=0;

for(int i = 0; i < N - 1; i++)
{
     t0 = 0.0 + (double)i * 0.01;
     t = t0 + 0.01;
     y=dy;

     try
     {
          integrator.integrate(ODE, t0, y, t, dy);
     }

catch(DerivativeException derivativeException) { }
catch(IntegratorException integratorException) { }
System.out.println(y[0]+"\n");
```

For the above code to work another class file needs to be created. In the example this is called OrdinaryDifferentialEquations. This class file is what describes the differential equation which we want the Runge-Kutta integrator

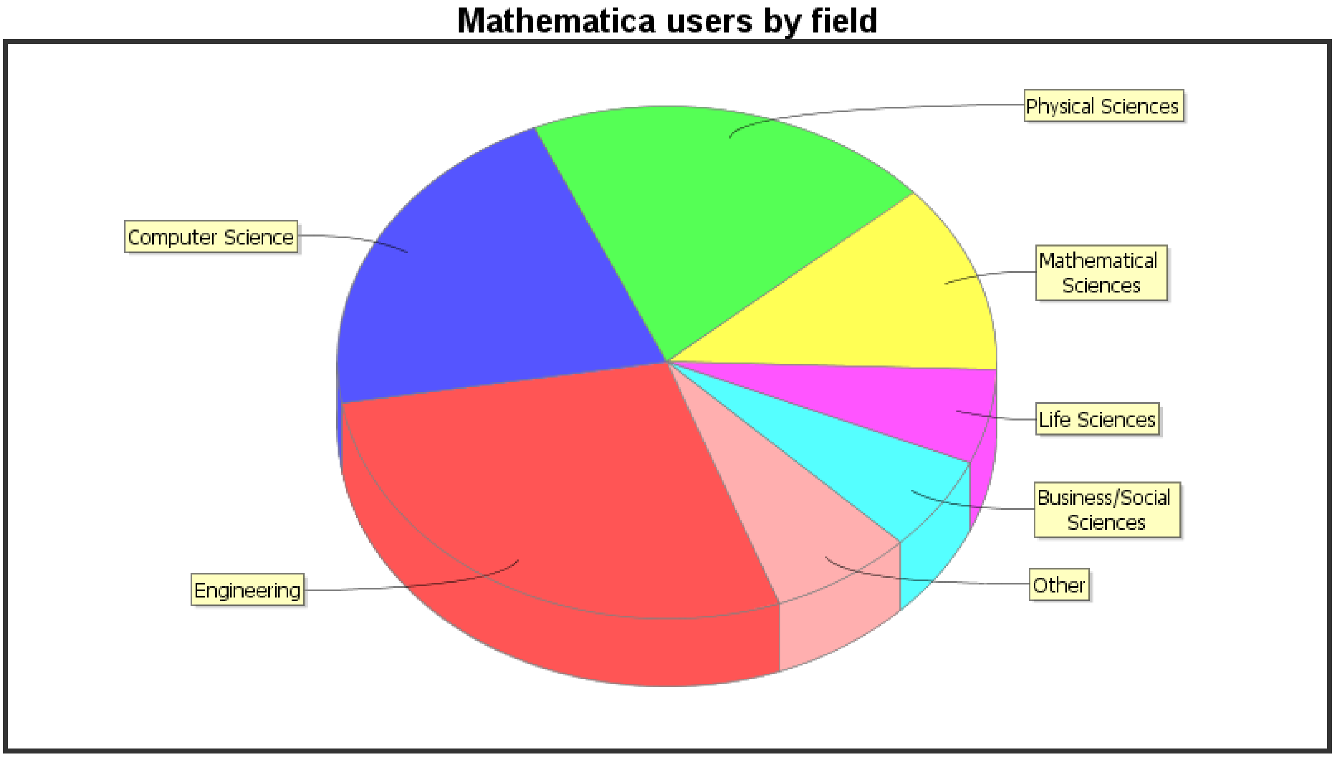


Figure 2: An example pie chart created using the JFreeChart library.

to solve. This part of the code is pretty straight forward and follows the same procedure as the MatLab equivalent function. The full example can by copied and pasted and only a small part needs to be changed in order for it to work. The part of the class that needs to be changed is the computeDerivatives method. This is where you program the differential equation. For our example we only use a first order differential equation, however higher order systems can be solved as well. For higher order systems they must be written as a set of first order equations. The getDimension method must return the correct size of the system to be solved. For this example a size of one. The below code is the computeDerivatives method.

```
public void computeDerivatives(double t, double y[], double yDot[])
{
        double A;
        double H;
        A = 10+4*(Math.sin(t));
        H = 2*Math.sqrt(y[0]);
        yDot[0] = A-H;
}
```

6. PUTTING THE PIECES TOGETHER

The final part of this project is to combine all the pieces together to create a interactive Java appet which simulates and graphs the control system. Writing the Java applet is fairly straight-forward. The graphical user interface can be customized to suit the system. The example for this paper uses four text fields for the user to input the initial conditions and one text field for the length of the simulation. The used can define these parameters and then run the simulation. A single button is used to recalculate and plot the two graphs. Simple Grid-Layout and Border-Layout commands were used. These are well documented in any Java programming resource.

The heart of the program is the Runge-Kutta function. The use of this is very similar to the ODE45 function in MatLab with some minor differences discussed earlier. It is also very well documented by Apache Commons. It is used by calling the integrator.integrate function and passing the needed variables. The example for this paper uses a simple ball and beam system which is described in the function BallAndBeam(). This function evaluates the right side of the differential equation and is nearly identical to the function that is called by the ODE45 function in MatLab. The DormandPrince54Integrator is used to solves the differential equations. After which the current data point is added to the graph dataset. This process repeats until the defined final time is reached.

The actual graphing is handled by the the JFreeChart function ChartFactory.createXYLineChart. This function creates the graph objects and defines aspects like the title and axis labels. As the data is calculated it is added to a dataset defined by one of the JFreeChart methods. A dataset can contain one or more series of data. The dataset is then added to the chart object. Once complete the chart can be displayed. JFreeChart handles the remaining code to properly draw the charts, adjust the axes, and various other tasks needed to get the data to the screen. This allows the programmer to focus on generating the data and not how to display it. JFreeChart offers many ways to manipulate the plots if further customization is required. Additionally, the example for this paper updates the graphs as the data is added so the user can watch as the plots are created.

7. DISPLAYING ON A WEBSITE

Once the applet is created it can be run locally by loading the applet. However, applets are uniquely useful for distribution. This is because an applet can be embedded in a web page and loaded by any person who visits the website without downloading or installing any software. A very simple html page needs to be created to load the the applet. This can be added to an existing page or a new page can be created dedicated to displaying the applet. The following code should be added to the html page:

```
<APPLET
   CODE="applet.class" WIDTH=900 HEIGHT=600
   ARCHIVE="jcommon-1.0.16.jar, jfreechart-1.0.13.jar,
   commons-math-1.2.jar">
</APPLET>
```

This html page and all the .class and .jar files listed in the above code should then be upload to the server which hosts the website. Remember that the .class file is the complied version of the Java code and the .jar files are the libraries which were downloaded. These files should all be placed in the same folder and specific care should be taken to make sure the files names match exactly. While running the files locally on a windows machine the capitalization of the file names is not critical because windows will recognize Test.jar the same as test.jar However, when loaded on to the server the capitalization of the file names must be identical. After the files are uploaded loading the html page with the applet code embedded will load the Java applet.

8. CONCLUSION

With these tools an interactive simulation of control systems can be displayed on a website. While Java programming is more difficult then the more common MatLab simulation program, it offers distribution benefits that MatLab does not. With the addition of JFreeChart to handle the plotting of data, and Apache's Mathematics library to handle the solving of ordinary differential equations programming controls simulations in Java is much easier. With a very simple user interface to change the initial condition and recalculate the simulation a interactive learning tool can be created. Because Java is a common part of Internet browsing these interactive simulations can be made available to anyone who has access to the Internet.

APPENDIX

A. ADDITIONAL RESOURCES

Here are some additional resources that might be helpful when creating Java applets. Additionally, the source code for the example ball any beam system is included.

- NetBeans IDE
- http://netbeans.org/index.html
- NetBeans IDE Java Quick Start Tutorial
- http://netbeans.org/kb/docs/java/quickstart.html
- JFreeChart
- http://www.jfree.org/jfreechart/
- Java Graphs and Charts
- http://www.informit.com/guides/printerfriendly.aspx?g=java&seqNum=74
- Chart a new course with JFreeChart
- http://www.javaworld.com/javaworld/jw-12-2002/jw-1227-opensourceprofile.html
- Commons-Math: Apache Commons Mathematics Library
- http://commons.apache.org/math/
- Class DormandPrince54Integrator
- http://commons.apache.org/math/apidocs/org/apache/commons/math/ode/nonstiff/DormandPrince54In