Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Educational Games in Control ⋆

Ulrich Münz *, Peter Schumm *, Frank Allgöwer *

* Institute for Systems Theory and Automatic Control,
University of Stuttgart, Germany
e-mail: {muenz,schumm,allgower}@ist.uni-stuttgart.de

Abstract: Basic control courses are often attended by students from many different study programs. Many of these students are mainly interested in solving practical problems, whereas the lecturer usually aims at teaching profound knowledge of the analysis and control of dynamical systems. This gap between the application-oriented expectation of the learner and the theory-focused material chosen by the lecturer may end up in a considerable demotivation of the students, which in turn lowers their learning performance. One way to close this gap and to increase both motivation and learning performance are educational games. This is shown in this paper exemplarily for two educational games that were introduced in a basic automatic control course at the University of Stuttgart. Both games are presented in detail and compared based on the course evaluations and feedback from our students.

Keywords: Control education, educational games, e-learning.

1. INTRODUCTION

Lecturers in control often face the problem how to teach complex, theoretical material to students who are mainly interested in solving practical control problems. In order to address this problem, university lecturers typically put a lot of effort into the improvement of the way they teach. Especially the integration of computers and new information technologies offer a large number of possibilities. In this respect, new media helps to illustrate the structure of complex systems, virtual and remote labs help to gain practical experience anytime and anywhere, see for example [Sánchez et al., 2004, 2002, Johansson et al., 1998] and references therein.

In addition to providing good teaching materials and learning facilities, adequate motivation is essential for learning. A promising approach to achieve this necessary motivation is the use of computer-games [Foreman et al., 2004, Zyda, 2007]. This concept to catch the attention and enforce the motivation of the learner has a longer tradition in games for school children (K-12 education). Different studies show that, especially in mathematics and physics, the use of games helps to improve the learning progress [Randel et al., 1992, Mayo, 2007b]. With computers being able to simulate more complex models, more and more educational games are also developed for learning at university level, see for example Becker [2001], Reese [2000], Foss and Eikaas [2006], Haugom et al. [2006], Hennessey and Kumar [2006], Kelly et al. [2007]. Latest developments in this area are presented in two recent special issues on educational games [Mayo, 2007a, Mayer and van der Voort, 2006]. An *Educational Game Model* as well as an overview on educational theories related to games are presented in Amory and Seagram [2003].

The advantages of computer-games in education are easy to see. It is normally a difficult job to convince students to do some extra preparations for the course in their spare time. On the other hand, a lot of people spend voluntarily hours in computer-games for training some new moves, learning the special features of a weapon, or developing better strategies for producing some virtual goods. The behavior of these players is very similar to what good students should do. They keep on trying to solve a given problem using their already gained knowledge combined with new strategies. Having passed one level, they are very keen on facing the new challenges of the next level. At the same time, a game can behave like a good instructor. It continuously monitors the progress and gives immediate feedback to the player. In doing so, the player can progress at his own speed according to his own abilities. Another positive side effect is that gamers tend to exchange their experiences. They talk about their strategies and they try to be better than their "competitors".

These are only a few reasons why we think that computer-games are appropriate for enhancing the motivation and learning progress of students. Mayo [2007b] summarizes different didactical features that are inherently used in most computer-games, e.g., interactivity, experimental learning, continuous feedback, and several levels with increasing difficulty. Learning science has shown that these concepts help to improve learning performance. An important aspect for designing educational games is to maintain these didactical features.

This paper shows how two educational games have been integrated in a basic automatic control course at the undergraduate university level with more than 200 students every year. Both games have been developed with very different main objectives. The first game is called *sub-*

⋆ U. Münz's work was financially supported by The MathWorks Foundation and by the Priority Programme 1305 "Control Theory of Digitally Networked Dynamical Systems" of the German Research Foundation.

marine [IST, 2007c] and aims mainly at motivating the students. Moreover, it provides an intuitive user interface that invites the student to "turn the knobs". This game has been presented in a recent special issue on e-learning [Münz et al., 2007]. The name of the second game is *spaceball* [IST, 2007b]. Its main objective is to motivate students to gain deeper understanding of advanced control problems and even of nonlinear dynamics. Therefore, we provide the students with the source code of the game, such that they can implement new features themselves. The main contribution of this paper is the presentation and comparison of the two different approaches of educational games as well as to report about our classroom experiences for both games.

The paper is organized as follows: The course is briefly described in Section 2. Both games are explained in detail in Section 3. The comparison of both games based on student evaluations is presented in Section 4 before we conclude the paper in Section 5.

2. AN AUTOMATIC CONTROL COURSE AT THE UNIVERSITY OF STUTTGART

The educational games presented in this paper are used in our course *Automatic Control 1* at the University of Stuttgart. It is part of the curriculum of several study programs, such as engineering cybernetics, process engineering, or mechanical engineering. Most of the about 200 attendees are third year students. The course consists of 45 hours of lectures and 15 hours of exercises during one semester, i.e. 4 hours per week over a period of 15 weeks which corresponds to 6 ECTS (European Credit Transfer System). In the lectures, new material is introduced to the students, while during the exercises common control problems and typical solution strategies are demonstrated.

This course covers the basics of analysis and control of linear time-invariant systems in the time and frequency domain, e.g. state-space representation and pole placement, Laplace transform and transfer function, Bode diagram, Nyquist stability criterion, and PID controller design. Clearly, the course content is typical for many basic control courses. Therefore, we hope that our experience in applying educational games for motivating students and improving their learning process is transferable and might be of help to other lecturers.

Apart from the educational games described below, we use further e-learning and conventional training modules to enhance our teaching in this course. Exemplarily, we mention online exercises for loop-shaping controller design or a mini-quiz given to the students on a weekly basis. More details can be found on the course homepage [IST, 2007a] and in a recent publication on our e-learning activities for this course [Schweickhardt et al., 2006].

3. EDUCATIONAL GAMES

The aim of both educational games is to control the movement of one or several vehicles such that they follow a given trajectory. The horizontal speed is constant whereas the vertical speed can be manipulated either by hand or by automatic controllers. The underlying model for the vertical position of the vehicle is a double integrator.

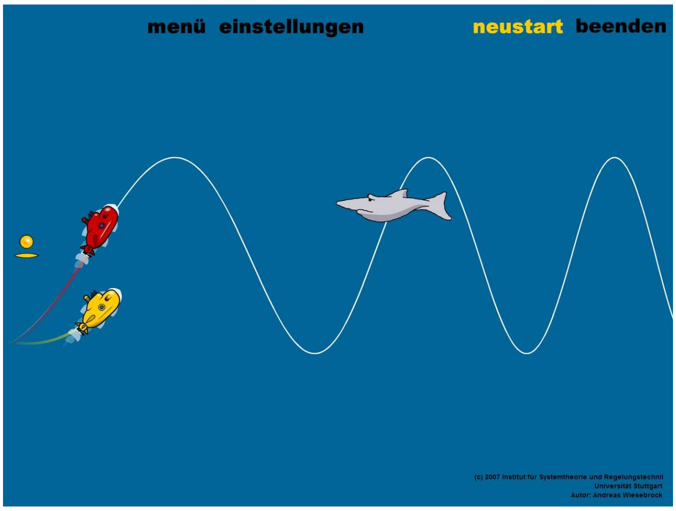

Fig. 1. Screenshot of our *submarine* game: Two submarines following a sweep sinus reference signal. The shark is just a gimmick to give the game a nice appearance.

This simple model structure was chosen for several reasons: First, all students can easily understand the dynamics of the vehicle from Newton's second law of motion. Second, the dynamics is unstable. Therefore, it is quite difficult to steer the vehicle by hand. Hence, the benefit of using feedback control becomes clear to the students as soon as they play the game for the first time. Third, the model is simple enough to solve basic control problems like pole placement with pencil and paper. Summarizing, these features are important for our didactical aim: to increase the motivation and learning performance of the students. The control task is related to standard problems for engineers as well as challenging and solvable in reasonable time.

The two different games are now explained in Section 3.1 and 3.2. A comparison is given in Section 4.

3.1 Submarine Game

The first game presented in this paper deals with a submarine that has to be maneuvered through the water [IST, 2007c], see Figure 1. The *submarine* game exhibits various modes. These modes correspond to the different parts of the automatic control course. Thereby, the game appears as a recurring theme in the lecture.

At the beginning of the course, *submarine* is introduced as a game: it has to be maneuvered by hand. It is rather difficult to steer the submarine by hand because it is unstable. Thus, the advantage of automatic control becomes obvious as soon as the lecturer types in a suitable controller. At that time, the students do not know how to design a controller, but they see immediately why the content of this course might be useful for them. This interest of the students in the course content is a good starting point for a fruitful learning process.

Whenever a new controller design technique is introduced in the course, the students are asked to apply this controller to the *submarine* game. Thereby, they use an example they are already familiar with. In particular, the following modes are implemented:

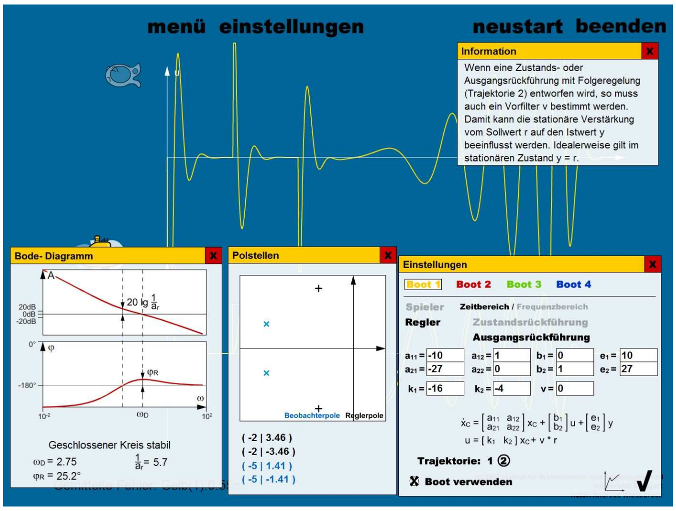

Fig. 2. Screenshot with controller parameter menu for an output feedback controller with state observer (right) in combination with the pole-zero-plot (middle) and the Bode-plot (left) to analyze the implemented controller.

- First, the students are asked to design a state feedback controller such that the poles of the closed-loop system are at specific points in the complex plane. For the evaluation of the controller, a pole-zero-plot (see Figure 2) shows the achieved closed loop characteristics. Thereby, the students learn how to read the pole-zero-plot and how they can change the poles of the closed loop with a state-feedback controller. Moreover, the connection between the dynamic behavior and the pole location of a system becomes clear.
- The next mode requires first the design of a state observer with certain error dynamics. The observed state as well as the real state are shown with different submarines in the game. Hence, the students can see how the observed state and the real state converge towards each other if the observer is well designed. The poles of the error dynamics are again shown in the pole-zero-plot. In a second step, a state observer and a state feedback controller have to be designed together. Thereby, the students experience the separation principle.
- The third mode deals with a PID controller design based on loop-shaping. Just like in real-life problems, the closed loop system has to satisfy certain objectives like stability, noise rejection, and robustness. Loop-shaping requires a profound understanding of the Bode-plot, which is also displayed in the game, see Figure 2. Hence, the students get used to this important tool when playing in this mode.

Finally, the expert mode offers the possibility to run up to four “competing” submarines in parallel with different controllers. Thereby, the students can compare the different controllers on a practical example. This is a good example for the final lecture of our course that wraps up the most important material covered in the class.

For all modes, the student can choose between two basic control objectives that are modeled by two different reference trajectories: The first trajectory is a constant value and corresponds to the stabilization of a set-point. The second trajectory is a sequence of steps and a sweep-sine-function, see Figure 1. This represents a tracking control problem.

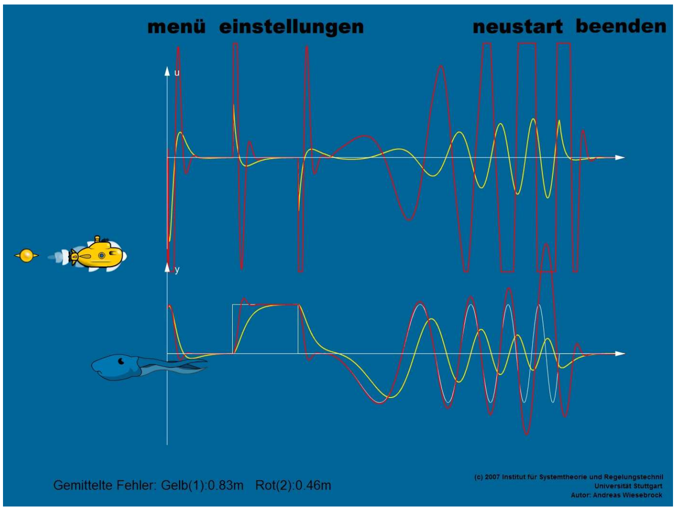

Fig. 3. Screenshot of the presentation of the input and output signals after a single run with two submarines.

Once all controller parameters and the desired trajectory are implemented, the user can press the start button and a simulation of the submarine with the implemented controller is shown. In the hand operated mode, the students can insert the lifting force using a scroll bar. In the expert mode, the reference signal for the closed loop is either given by the desired trajectory or by hand. Usually, the tracking control problem is solved much better with a stabilizing controller and manual reference command. This option is particularly important to maintain the interactivity of the game, i.e., the game is not running autonomously like virtual experiments. The game always runs for a couple of seconds. At the end of each run, the input and output signals of the controlled and hand-operated submarines are shown, see Figure 3. The students are now able to study and interpret the dynamic behavior of the controlled submarine. Typical effects like steady state offset, settling time, overshoot, or input saturation can be easily observed. All these characteristics are also reflected in the system analysis tools pole-zero-plot and Bode-plot. Hence, the students can directly see the connection between these diagrams and the actual system behavior.

As an additional incentive for the students to improve their controllers, we implemented a course-wide web-based highscore list. In the expert mode, the students can send their obtained results to the server of our institute. Thus, they can compare the achieved performance with their classmates. In particular, the mean square error of the control error between the desired and the achieved trajectory is calculated as performance measure. The highscore list is separated for different types of controllers, i.e., state-feedback, PID-controller, etc., in order to get a fair picture.

Since the main objective of the *submarine* game is motivation, the graphical appearance plays an important role in attracting the students attention. Hence, Macromedia Flash [Adobe, 2007] was chosen as technological platform. Graphics and animations can be implemented easily, as for example additional gimmicks like the fish in Figures 1.

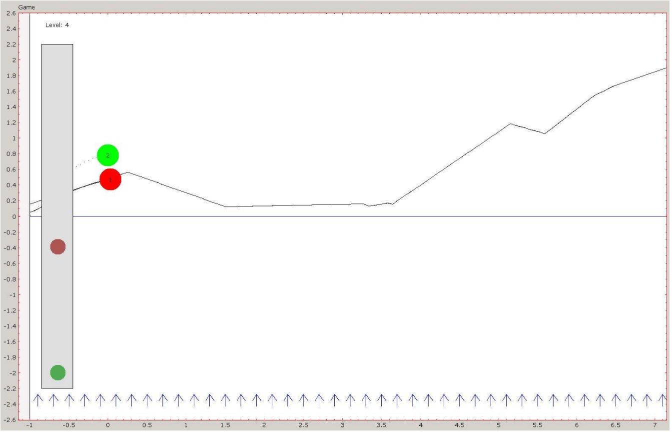

Fig. 4. Screenshot of our *spaceball* game: Two spaceballs following a random walk reference signal. The blue arrows at the bottom indicate the solar wind.

The needed mathematical functions are simple enough to be implemented with Flash. In fact, only a routine for the simulation of a set of linear differential equations had to be programmed. Moreover, Macromedia Flash programs can either be embedded in HTML-code or exported as executable Windows-files. Therefore, the program runs both autonomously or in a standard browser, but there is no need for any further programs.

3.2 Spaceball Game

The second game is about an object called *spaceball* that flies through the universe [IST, 2007b]. The *spaceball* game offers a more difficult environment compared to the *submarine* game. It is divided in five different levels with increasing degree of difficulty.

In the first level, the spaceball only has to follow a given trajectory. Altogether, there are five possible trajectories: two sinusoidal and two rectangular functions as well as a random signal, see Figure 4. In the second and third level, the spaceball randomly changes its weight with increasing probability, indicated on the screen by a larger or smaller ball. This strongly affects the dynamics of the spaceball which in turn makes it more difficult to track the given trajectory. In the fourth and fifth level, a solar wind with increasing intensity interferes the movement of the spaceball. It acts as a disturbance on the dynamical system and is represented in the game by blue arrows on the upper or lower end of the screen, see Figure 4.

In order to steer the spaceball through the different levels, the player has the choice between different controllers. First of all, the spaceball can be steered by hand of course. As for the submarine, this is a quite difficult task because the system is unstable. Apart from that, the spaceball can be controlled using P, PI, PD, or PID controllers. In contrast to the *submarine* game, there are no Bode-plot or similar analysis tools available for the design. The controller can only be improved by repetitive simulation of the closed loop. At the end of each run, the resulting input, output, and disturbance signals are shown, see Figure 5. Here, the student can study the disturbance rejection and robustness properties of the closed loop.

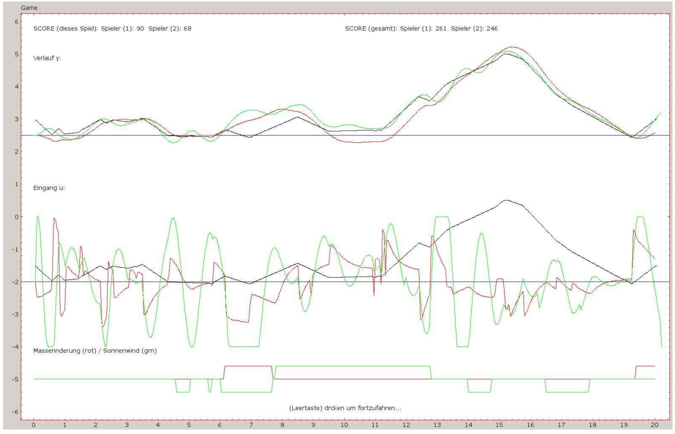

Fig. 5. Screenshot of the presentation of the input, output, and disturbance signals after a single run with two spaceballs.

In order to offer the students the possibility to see and exchange the source code of this game, it was developed as an application running on the Sysquake environment [Calerga, 2006]. Sysquake is a simulation and visualization tool, based on a Matlab-like programming language. A "limited edition" (LE) is freely available. The implementation and solution of differential equations is straightforward in Sysquake. However, the implementation of a graphical user interface requires some more work than in Flash, even though the graphics are much simpler in Sysquake. The implementational effort for both games was comparable.

The main idea is that our students download this free edition from the Calerga webpage and a basic source code of the *spaceball* game from our homepage [IST, 2007b]. Then, the students can play the game by hand or using one of the different controllers. Moreover, the students have the possibility to change the source code directly. This way, they could include additional analysis tools like a Bode-diagram. Moreover, the dynamics of the spaceball could be changed, even into nonlinear dynamics. Alternatively, other control structures from the lecture could be implemented. Hence, the students have even the opportunity to study more advanced control topics in a playful way.

4. EXPERIENCES AND COMPARISON

We introduced the *submarine* game in our automatic control lecture with the intention to increase the general motivation and to improve the learning performance. In order to study the effectiveness of the game, an evaluation with more than 150 students took place in February 2007.

The evaluation showed that about 97.5% of the students in the class used the *submarine* game during the lecture period. Note that the usage of the *submarine* game is on a purely voluntary basis. The students were never urged to solve homework problems or exercises using this game. We obviously managed to motivate almost all our students to use the game at least every now and then.

Moreover, we asked our students if the game helped them to understand the course material. On a scale from 1 (*I totally agree*) to 7 (*I totally disagree*), about two thirds

Table 1. Comparison of the educational games *submarine* and *spaceball*.

Criterion	*submarine*	*spaceball*
Teaching related properties		
Target audience	beginners	advanced students
Main educational aim	motivation and learning progress	deepening and extending understanding
Control related properties		
Implemented controllers	state-feedback, output-feedback with observer, PID	PID
Implemented graphical analysis tools	Bode-diagram, pole-zero plot	none
Additional control problems (not covered in the lecture)	none	uncertainties (change of weight of vehicle), disturbances (solar wind)
Extension to more difficult dynamics and controllers	rather difficult	easy to implement in source file
Reference signal	constant, steps and sweep-sinus	sinusoidal, rectangular, random
Implementation related properties		
Programming language	Flash	Sysquake
Required additional software	web-browser	Sysquake LE (freely available)
Graphical appearance	fancy graphics with gimmicks easy to implement	simple graphics
Implementational effort for the game	medium	medium
Course-wide highscore list	yes	no (difficult to implement)
Intuitive user interface	yes	yes, knowledge of Sysquake/Matlab helpful

of our students marked 1 or 2, indicating that they find the submarine game very helpful for their learing progress. Only about 5% marked 6 or 7. Apart from the fact that the game motivated the students to solve control problems, they clearly also have the feeling that their learning progress improved by the use of this game.

The fancy appearance using Macromedia Flash [Adobe, 2007] was supposed to attract the students' attention. Yet, some of them say that the appearance is not the main reason for trying out the game. The introduction of a course-wide highscore list was a great success. Only one week after the introduction of the submarine game in our course, the list contained more than 400 entries.

On the other hand, our experiences with the *spaceball* game show that the idea of an open source code is more suitable for advanced students. Many students of our course struggle with the controller design. Hence, they are not looking for more challenging problems like model uncertainties, disturbance rejection, or even nonlinear dynamics. On the contrary, the majority prefers a ready-made game that provides all analysis and synthesis tools such that they only have to insert the proper controller parameters. For these students, the *submarine* game is perfectly fine. The *spaceball* game is mainly played by the more advanced students. A detailed comparison of the two educational games is presented in Table 4.

We also see from the comments of our students that a continuation of the control games in more sophisticated courses is highly appreciated. Therefore, the platform like Sysquake is very promising. When the students focus on more sophisticated analysis and design problems, they cannot solve them by typing parameters in ready-made masks. In fact, they need to learn how to implement such problems in standard software like Matlab.

In additional remarks of the evaluation, our students stress the fact that our educational games show the link between control theory and "real world problems" even though the dynamics are very simple. This indicates that educational games can help to solve the principal dilemma of many control courses: How to close the gap between profound theory and practical applications? Yet, it is important for a successful implementation of educational games to preserve important didactical features of games as described in the Introduction. For example, we achieved *interactivity* by introducing a manual reference command for the closed loop system.

5. CONCLUSIONS

Motivation is a cornerstone for learning at any level of education. In university level control education, the gap between the application-oriented expectations of the learner and the theory-focused lecture content may cause considerable demotivation of a number of students. Educational games are one way to close this gap. In this paper, we presented two educational games that were introduced in a basic automatic control course at the University of Stuttgart.

The *submarine* game achieves most of our aims: The students are motivated by a practical example, which shows them the difficulties of controlling a system and the advantages of automatic control. Moreover, they increase their knowledge on the course matter in a playful way using different controller design techniques and standard tools for system analysis. The use of games like *spaceball*, where the students can implement advanced algorithms in the source code, requires at least a basic knowledge of control. Hence, we see this kind of games suitable for advanced control education.

Overall, our experience shows clear benefit from the use of educational games in control education. Along this line goes our current project to introduce *spaceball* game in our advanced classes.

ACKNOWLEDGEMENTS

We would like to thank all colleagues and students who helped designing and implementing the educational games:

Tobias Schweickhardt, Maarten Haubrich, Andreas Wiesebrock, and Johannes Eck.

REFERENCES

Adobe. Macromedia Studio homepage. http://www.adobe.com/products/studio/, 2007.

A. Amory and R. Seagram. Educational game models: Conceptualization and evaluation. *South African Journal of Higher Education*, 17(2):206–217, 2003.

K. Becker. Teaching with games: the minesweeper and asteroids experience. *Journal of Computing Sciences in Colleges*, 17(2):23–33, 2001.

Calerga. SysQuake homepage. http://www.calerga.com/products/Sysquake, 2006.

J. Foreman, J. P. Gee, J. C. Herz, R. Hinrichs, M. Prensky, and B. Sawyer. Game-Based Learning: How to Delight and Instruct in the 21st Century. *EDUCAUSE Review*, 39(5):50–66, 2004.

B. A. Foss and T. I. Eikaas. Game play in engineering education — concept and experimental results. *International Journal of Engineering Education*, 22(5):1043–1052, 2006.

R. Haugom, O. K. Solbjørg, K. Y. Pettersen, and T. I. Eikaas. A simulation game for nonlinear control theory education. In *Proceedings of the 6th IFAC Symposium on Advances in Control Education*, Madrid, Spain, 2006.

M. Hennessey and S. Kumar. Integrated graphical game and simulation-type problem based learning in kinematics. *International Journal of Mechanical Engineering Education*, 34(3):220–232, 2006.

IST. Institute for Systems Theory and Automatic Control, University of Stuttgart: Automatic Control 1 homepage. http://www.ist.uni-stuttgart.de/education/courses/RTI/, 2007a.

IST. Institute for Systems Theory and Automatic Control, University of Stuttgart: Space-Ball homepage. http://www.ist.uni-stuttgart.de/education/elearning/rtgame/, 2007b.

IST. Institute for Systems Theory and Automatic Control, University of Stuttgart: Submarine-Game homepage. http://www.ist.uni-stuttgart.de/education/courses/RTI/submarine.shtml, 2007c.

M. Johansson, M. Gäfvert, and K.J. Åström. Interactive tools for education in automatic control. *IEEE Control Systems Magazine*, 18(3):33–40, 1998.

H. Kelly, K. Howell, E. Glinert, L. Holding, C. Swain, A. Burrowbridge, and M. Roper. How to build serious games. *Communications of the ACM*, 50(7):44–49, 2007.

I. Mayer and H. van der Voort. Special issue on Games for Design and Engineering Education, *Journal of Design Research*, 5(2):155-298, 2006.

M. J. Mayo. Special issue on Creating a Science of Games, *Communications of the ACM*, 50(7):26–73, 2007a.

M. J. Mayo. Games for science and engineering education. *Communications of the ACM*, 50(7):30–35, 2007b.

U. Münz, P. Schumm, A. Wiesebrock, and F. Allgöwer. Motivation and learning progress through educational games. *IEEE Transactions on Industrial Electronics, Special Section on "e-Learning and Remote Laboratories within Engineering Education"*, 54(6):3141–3144, 2007.

J. M. Randel, B. A. Morris, C. D. Wetzel, and B. V. Whitehill. The effectiveness of games for educational purposes: A review of recent research. *Simulation & Gaming*, 23(3):261–276, 1992.

D. S. Reese. Using multiplayer games to teach interprocess communication mechanisms. *SIGCSE Bulletin*, 32(4): 45–47, 2000.

J. Sánchez, S. Dormido, R. Pastor, and F. Morilla. A Java/Matlab-based environment for remote control systems laboratories: Illustrated with and inverted pendulum. *IEEE Transactions on Education*, 47(3):321–329, 2004.

J. Sánchez, F. Morilla, S. Dormido, J. Aranda, and Ruipérez. Virtual and remote control labs using Java: A qualitative approach. *IEEE Control Systems Magazine*, 22(2):8–20, 2002.

T. Schweickhardt, P. Schumm, U. Münz, and F. Allgöwer. Integration of e-learning modules in automatic control education. In *Proceedings of the 6th IFAC Conference on Advances in Control Education*, Madrid, Spain, 2006.

M. Zyda. Creating a science of games. *Communications of the ACM*, 50(7):26–29, 2007.