2004-01-01

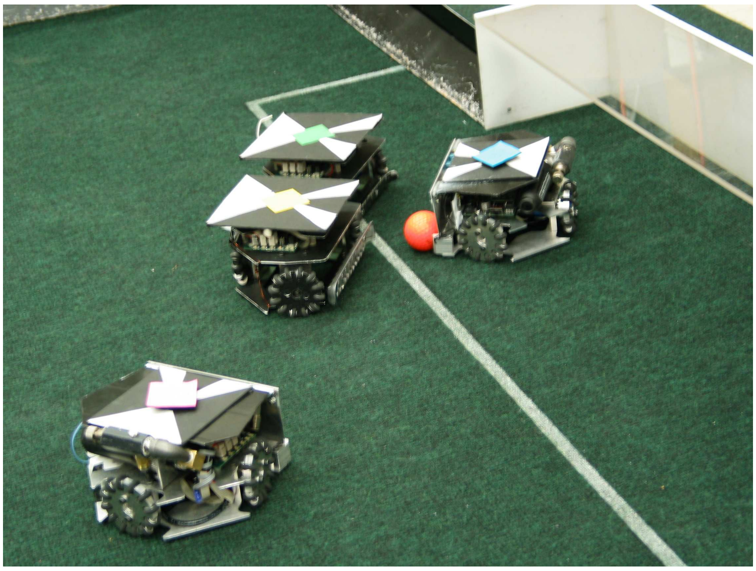
Motivating Undergraduate Feedback Control Through Robot Soccer

Randal Beard
beard@byu.edu

Original Publication Citation

Randal W. Beard and James K. Archibald and Steven A. Olson, "Robot Soccer as a Culminating Design Project for Undergraduates," American Control Conference, Anchorage, Alaska, May, 2002, p. 1086-1091.

BYU ScholarsArchive Citation



MOTIVATING UNDERGRADUATE FEEDBACK CONTROL THROUGH ROBOT SOCCER

Randal Beard

Department of Electrical and Computer Engineering

Brigham Young University,

Provo, UT 84602

beard@ee.byu.edu

One of the challenges in control engineering education is to motivate students to invest the effort necessary to understand the theoretical foundations of feedback control. We have found that many students are inherently interested in robotics and are attracted to control engineering when they understand that it plays a central role in robotic systems. To capitalize on this appeal, we have established a one semester robot soccer senior design course that uses the undergraduate feedback control course to introduce the prerequisite material. As a consequence, the undergraduate feedback control course and its lab have been redesigned to focus on mobile robots with a particular emphasis on techniques helpful in robot soccer.

Robot soccer requires several applications of feedback control concepts. These include low level velocity control of the robots, trajectory tracking, trajectory generation, state estimation, and ball prediction. To enable the students to implement these concepts, several topics that are not traditionally taught in an undergraduate feedback control course must be introduced. These topics include mathematical models for mobile robots, feedback linearization, Kalman filtering, and path planning. This article gives a brief overview of our approach to introducing these topics at a level appropriate for undergraduates in electrical, computer, and mechanical engineering. We assume that the students have taken introductory courses in signals and systems and linear algebra.

Overview of Curriculum

Table 1 gives an overview of the topics covered in the one semester, senior level, undergraduate feedback control course taught in the Electrical and Computer Engineering Department at BYU. Lecture material and reading assignments for weeks 1–3, 5–9, and 13–15 follow a standard feedback control textbook, for example [1]. Topics are introduced in an order that facilitates the laboratory assignments. Students are not required to implement frequency domain techniques in the lab, therefore these subjects are postponed to the last three weeks of the course.

Week	Lecture Topic	Lab Assignment
1	Models of DC motors and mobile robots.	
2	Time domain specifications and their relationship to pole and zero location.	Simple programming assignments using the microcontroller board on the robots.
3	PID control, steady state tracking.	Design and implement PID controller for DC motor.
4	Mobile robot velocity control and trajectory tracking.	Design and implement velocity controller for mobile robot in robot frame using wheel encoders.
5	Root locus.	
6	Review of linear algebra.	Design and implement trajectory tracking controller for mobile robot using vision feedback.
7	State space methods.	
8	Pole placement / LQR.	Design and implement controllers for the following modes: straight-line defense, goto point, turn to angle.
9	Observers.	
10	Kalman filter.	Design and implement Kalman filter to predict the kinematic states of the robot using camera measurements.
11	Extended Kalman filter and ball prediction.	Design and implement extended Kalman filter to predict future positions of the ball.
12	Motion planning.	Design and implement simple waypoint path planner, trajectory generator, and trajectory tracker.
13	Frequency response, Bode plots.	
14	Stability margins.	
15	Loopshaping.	

Table 1: Course outline. Topics covered in the undergraduate feedback control course with their associated lab assignments.

The addition of new topics has necessitated that we eliminate subjects that are traditionally covered in this course. For example we do not teach the Routh-Hurwitz criteria and we devote only two lectures to root locus. In addition, our coverage of frequency domain design is significantly reduced with only two and a half weeks dedicated to this topic.

Feedback Control for Mobile Robots

Robot control is decomposed into two layers: low-level, high-bandwidth velocity control implemented on the robot and low-bandwidth trajectory tracking implemented at the base station. To accommodate velocity control, each robot is equipped with a motor control board that includes a 29.4 MHz Rabbit microcontroller, 512k FLASH, 512k SRAM, four quadrature encoder inputs, six PWM motor driver channels with 2 amp H-bridges, four low current PWM channels, four serial ports, a 5 Volt, 2 Amp regulated power source, and ten 12 bit analog inputs [2]. The low-level control architecture used for velocity control is shown in Figure 1 where v and ω are the speed and angular speed of the robot center of mass, Ω_ℓ and Ω_r are the angular speed of the left and right wheels respectively, $k_{*,*}$ are motor parameters, and the superscript 'c' denotes a commanded value. The matrix M translates v and ω to Ω_ℓ and Ω_r and is given by

$$M = \begin{pmatrix} \frac{1}{R} & \frac{b}{R} \\ \frac{1}{R} & -\frac{b}{R} \end{pmatrix},$$

where R is the radius of the wheels and b is the radius of the robot.

The inner PD loops shown in Figure 1, regulate the wheel speeds and are tuned to be robust with respect to $k_{*,*}$. The outer PI loop regulates the robot speeds and is tuned to robust with respect to R and b. The feedback loops shown in Figure 1 run at approximately 120 Hz on the motor control board.

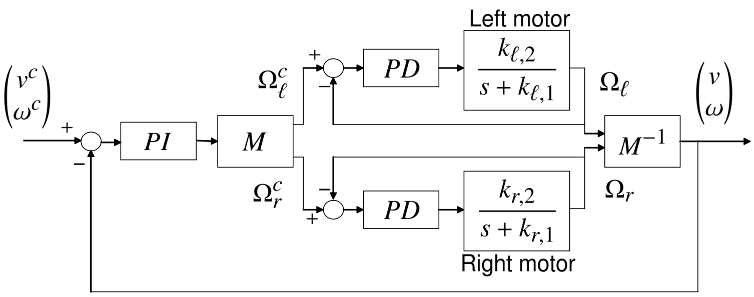


Figure 1: Low-level velocity control. The velocity controller for a two wheeled robot consists of PD loops for both the right and left motors and an outer PI loop to account for model mismatch.

Trajectory tracking is implemented on the base station and uses vision feedback at 30 Hz to correct for deviations from the desired trajectory. Using the velocity controller shown in Figure 1 the kinematic equations of motion are given by

$$\begin{pmatrix} \dot{r}_x & \dot{r}_y & \dot{\psi} \end{pmatrix}^T = \begin{pmatrix} v^c \cos\psi & v^c \sin\psi & \omega^c \end{pmatrix}^T,$$

where (r_x, r_y) is the inertial position of the center of mass of the robot and ψ is the heading direction.

As shown in Figure 2, we define the robot hand position $\mathbf{z}$ to be $\mathbf{z} = \begin{pmatrix} r_x & r_y \end{pmatrix}^T + L \begin{pmatrix} \cos\psi & \sin\psi \end{pmatrix}^T$. Differentiating $\mathbf{z}$ we obtain

$$\dot{\mathbf{z}} = \begin{pmatrix} \cos\psi & -L\sin\psi \\ \sin\psi & L\cos\psi \end{pmatrix} \begin{pmatrix} v^c \\ \omega^c \end{pmatrix} \triangleq R(\psi) \begin{pmatrix} v^c \\ \omega^c \end{pmatrix}.$$

Therefore, the feedback linearizing control

$$\begin{pmatrix} v^c & \omega^c \end{pmatrix}^T = R(\psi)^{-1} \left(\dot{\mathbf{z}}^r - \gamma \left(\mathbf{z} - \mathbf{z}^r\right)\right)$$

results in expontial convergence of the tracking error.

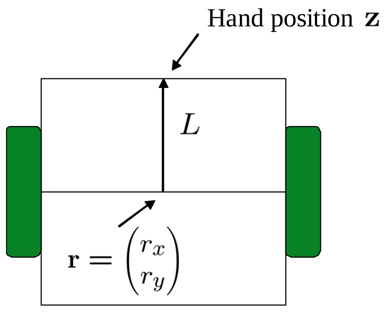


Figure 2: The hand position of the robot. To facilitate feedback linearization, the dynamics of the robot are expressed around a position $\mathbf{z}$ at the front of the robot.

State Estimation

We have found that a well tuned Kalman filter that estimates the robot position and heading are essential in robot soccer. In addition, an extended Kalman filter is an effective technique to estimate the current and future positions of the ball. Therefore, we have decided to introduce Kalman filtering in the undergraduate feedback control course.

We introduce Kalman filtering in seven steps. (1) The first step is a standard introduction to state estimation following, for example, the treatment given in [1]. (2) The second step is to introduce the notion of state estimation at discrete samples, given a continuous linear process. We also discuss how multiple sensors operating at different sample rates can be used to update the estimate. This occurs in robot soccer since the camera provides position estimates at 30 Hz and the wheel encoders provide velocity estimates at about 120 Hz. (3) The next step is to overview the probabilistic notions of mean and covariance, with a particular focus on developing the intuition behind the meaning of the covariance matrix. (4) We then define the problem of minimizing the mean and covariance of the estimation error as $t \to \infty$. (5) The Kalman filter equations are introduced as the solution to this problem. However, we do not derive the filter or prove any mathematical properties. The students are encouraged to enroll in the appropriate graduate level course for further explanation. The focus is on practical use and implementation of Kalman filters. (6) We then extend the idea to nonlinear systems and describe the extended Kalman filter (EKF). (7) Finally, several examples complete with Matlab implementation are presented. The examples include a single integrator, two wheel and three wheel mobile robots, and ball dynamics.

For a two-wheel mobile robot, we assume that the equations of motion are given by

$$\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} (v^c + \delta v)\cos(\psi) \\ (v^c + \delta v)\sin(\psi) \\ (\omega^c + \delta\omega) \end{pmatrix} = \begin{pmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v^c \\ \omega^c \end{pmatrix} + \begin{pmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \delta v \\ \delta\omega \end{pmatrix},$$

$$y = \begin{pmatrix} r_x & r_y & \psi \end{pmatrix}^T + \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}^T$$

where n_* is measurement noise and δv and $\delta\omega$ are the mismatch between commanded and actual velocity. We explain that to apply the EKF, these quantities are assumed to be zero mean Gaussian random variables. Students are required to implement an EKF for a mobile robot in the lab.

Students are also required to implement an EKF that estimates the position of the ball and predicts its future position at time T in the future. The dynamics of the ball are given by

$$\begin{pmatrix} \ddot{b}_x & \ddot{b}_y \end{pmatrix}^T = -\frac{\mu \begin{pmatrix} \dot{b}_x & \dot{b}_y \end{pmatrix}^T}{m\sqrt{\dot{b}_x^2 + \dot{b}_y^2}} + \frac{1}{m} \begin{pmatrix} \delta F_x & \delta F_y \end{pmatrix}^T,$$

where $(b_x, b_y)^T$ is the position of the ball, m is the mass, μ is the coefficient of friction, and $(\delta F_x, \delta F_y)$ are unmodeled forces acting on the ball. We assume that $(\delta F_x, \delta F_y)$ are zero mean Gaussian random variables and proceed as normal. Students find the ball prediction lab to be the most difficult lab assignment.

Path Planning

Our introduction to path planning focuses on the architecture shown in Figure 3 where our approach to trajectory tracking and the velocity control have been discussed in previous sections. One lecture is devoted to introducing a simple geometric technique for planning straight-line waypoint paths that avoid obstacles. The result of the waypoint path planner is a set of waypoints $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_M\}$, and a desired velocity v^d along the waypoint path.

The trajectory generator transforms $\mathcal{P}$ into a time-parameterized path for the hand position $\mathbf{z}(t) = (z_x(t), z_y(t))^T$. The approach that we introduce in class is outlined below.

1. Parameterize the straight-line segment between $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ by a non-dimensional parameter τ which ranges from 0 to 1 over the straight-line segment, that is, let $\mathbf{z}^r(t) = (1 - \tau(t))\,\mathbf{p}_i + \tau(t)\mathbf{p}_{i+1}$.
2. Derive a differential equation for τ that causes the waypoint path to be travelled at constant velocity v^d:
$$\dot{\tau} = \frac{v^d}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}.$$
3. Modify the differential equation to take into account tracking error:
$$\dot{\tau} = \frac{v^d}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|} \left(\frac{\text{error}_{\max} - \|\mathbf{z}(t) - \mathbf{z}^r(t)\|}{\text{error}_{\max}} \right).$$
4. The time-parameterized trajectory is created by integrating τ with respect to time and resetting as the trajectory transitions from one straight-line segment to the next.

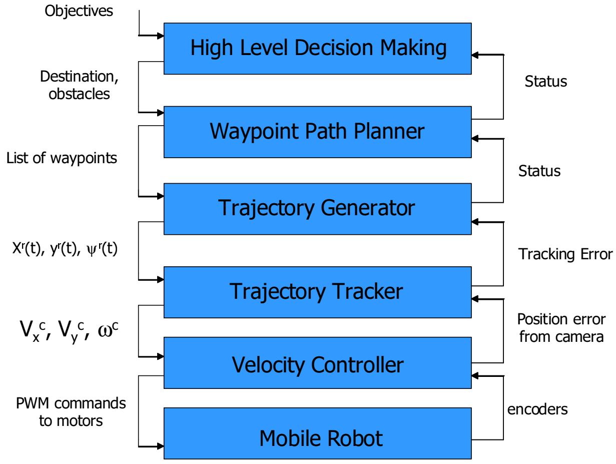


Figure 3: Software architecture for motion planning. Motion planning is decomposed into four hierarchical levels, each designed using feedback control techniques.

Discussion

The undergraduate feedback control course described in these notes is taught in the Fall Semester and is followed by the robot soccer senior design in the Winter semester. In the robot soccer course, teams of four to five students are required to design and build two robots and write the associated software [3]. The format is a two-on-two competition patterned after the Robocup standards [4]. Additional topics taught during the first month of the robot soccer course include basic ideas from computer vision and artificial intelligence.

Robot soccer provides a compelling introduction to feedback control for several reasons. First, the need for feedback is obvious. Second, when the control elements discussed in this note are implemented and well tuned, the performance benefits are dramatic. Finally, the application to robot soccer is fun and inherently motivating to many students.

After the introduction of robot soccer and the associated changes in the undergraduate feedback control course, enrollment in feedback control in the Electrical and Computer Engineering Department at BYU has doubled from approximately 25 students per year to 50 students per year. In addition, we have found that students seem to have a better understanding of why feedback control is important and seem to glimpse the excitement that exists in the field.

References

[1] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Addison Wesley, 2002.

[2] The Magicc Board, Magicc Technologies, http://www.magicctech.com/magiccboard.html.

[3] J. K. Archibald and R. W. Beard, "Robot soccer for undergraduate students," *IEEE Robotics and Automation Magazine*, vol. 11, no. 1, pp. 70–75, March 2004.

[4] "The robot world cup initiative," http://www.robocup.org.

Biography

Randal W. Beard received the B.S. degree in electrical engineering from the University of Utah, Salt Lake City, in 1991, the M.S. degree in electrical engineering in 1993, the M.S. degree in mathematics in 1994, and the Ph.D. degree in electrical engineering in 1995, all from Rensselaer Polytechnic Institute, Troy, N.Y. Since 1996, he has been with the Electrical and Computer Engineering Department at Brigham Young University, Provo, UT, where he is currently an associate professor. In 1997 and 1998, he was a Summer Faculty Fellow at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. He is currently an associate editor for IEEE Control Systems Magazine and the IEEE Control Systems Society Conference Editorial Board. His research interests include small unmanned air vehicles, coordinated control of multiple vehicle systems, and nonlinear control.