Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47661

A Small, Low-Cost Undergraduate Laboratory for the Study of Graph Theory and the Networked Control of Multi-Agent Systems

Dr. Erick Rodriguez-Seda, United States Naval Academy

Erick J. Rodríguez-Seda received a B.S. degree in electrical engineering from the University of Puerto Rico, Mayaguez, in 2004 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 2007 and 2011, respectively.

He is currently a Professor at the Department of Weapons, Robotics, and Control Engineering at the United States Naval Academy. From 2011 to 2013, he was a Postdoctoral Research Associate at the University of Texas at Dallas. His current research interests include networked control, collision avoidance, and control of cyber-physical systems.

A Small, Low-Cost Undergraduate Laboratory for the Study of Graph Theory and the Networked Control of Multi-Agent Systems

Erick J. Rodríguez-Seda
rodrigue@usna.edu
Department of Weapons, Robotics, and Control Engineering
United States Naval Academy

Introduction

Multi-Agent Systems (MASs) are nowadays used in multiple applications, ranging from autonomous vehicles and smart grids to social networks and financial markets. Future control engineers and roboticists will need to understand the complex behaviors of large-scale networked systems and the challenges that distributed and unreliable communication networks can bring. Yet, traditional undergraduate control engineering curricula lack hardware-based, hands-on experience with multi-agent systems, partly due to the cost, laboratory space, and time required to conduct experimental activities.

This paper presents a low-cost experimental undergraduate testbed that provides a practical, portable small platform for studying graph theory concepts and the control of MASs. The testbed consists of five individual rotatory 1-degree-of-freedom (DoF) links connected via a Controller Area Network (CAN) bus, in which students can explore concepts such as graph connectivity, consensus, stability, and network reliability. Students are assumed to have basic C/C++ programming and linear algebra skills.

Background and Motivation

The ABET-accredited Robotics and Control Engineering (RCE) Undergraduate Program at the United States Naval Academy (USNA) introduces students to the study, design, and use of automation with emphasis on control theory, sensors and actuators, programming, and mobile robotics. Several advanced elective courses are offered to junior and senior students, including computer vision, control of robotic manipulators, digital control, optimal control, design and control of mobile vehicles, biomechanics, data science, embedded systems, and control of MASs, among others. Students enrolled in the Control of MASs course learn about graph theory as a representation of MASs, connectivity maintenance, consensus, formation control, navigation via artificial potential field functions, and coverage control. A complete list of topics is chronologically ordered in Table 1. The course syllabus follows a structure of two 50-minute

Table 1: Control of Multi-Agent Systems Topics

Topics	Lecture Hours	Laboratory Hours
Intro to MASs	1	2*
Set Theory and Graph Theory	2	–
Adjacency and Connectivity	1	–
Agreement (Consensus) Protocol	4	2*
Implementation of Consensus Protocol	–	4†, ‡
Communication Networks	1	–
Agreement Protocol with Time Delays	1	2†, ‡
Connectivity Maintenance	1	–
Kinematic Control of Mobile Robots	1	2†
Dynamic Control of Mobile Robots	1	2†
Formation Control	4	4*
Lyapunov Stability	3	–
Artificial Potential Field Functions	3	4*
Velocity Obstacles	1	–
Coverage Control	2	2*
Behavior-based Robotics	1	2*
Final Project Activity	–	8†

* Simulation-based laboratory activity
† Hardware-based laboratory activity
‡ Hands-on laboratory activity with MAS testbed

lectures and 110-minute hands-on laboratory sessions per week for a total of 16 weeks. Before enrolling in the course, students are assumed to be familiar with C/C++ and MATLAB programming languages and with concepts of classical and modern control theory, such as stability of linear systems, transfer functions, state space representation, controllability, observability, and Proportional-Derivative-Integral control. It is also assumed that they have some basic knowledge and experience with linear algebra, numerical integration, and the synthesis of closed-loop control systems.

The RCE undergraduate program at the USNA places significant emphasis on including small-group hands-on laboratory activities in each course. The Control of MASs course already used Unmanned Ground Vehicles (UGVs) for the design, implementation, testing, and validation of tracking and obstacle avoidance control algorithms for mobile robotics. However, using the same hardware and laboratory for topics on graph theory and consensus provided some challenges. Each group of students would need to work with a MAS of at least four UGVs to observe the different effects and convergence properties of the agreement protocol under a wide range of communication graphs. The limited classroom space and the limited number of available UGVs did not allow small groups to work simultaneously. Furthermore, the control of multiple UGVs required, in general, more class time to set up experiments and collect data. Therefore, the course needed a laboratory testbed that was compact, easy to replicate, and relatively easy and fast to use. This need motivated the design and construction of the small, low-cost 5-agent testbed described in this document.

Related Work

For decades, the use of hands-on laboratory experience has been seen as an important part of any comprehensive undergraduate robotics and control engineering curriculum [1–3]. A recent survey among robotics instructors representing 67 different institutions of higher education found that 75% of all participants agreed that laboratory experience is an integral part of robotics education [4]. Examples of hands-on activities for advanced robotics and control engineering topics abound in the literature (refer to [5–7]). For example, the laboratory in [8, 9] introduces students to the topic of cyber security in robotics by exposing them to the vulnerabilities of networked control systems. Students can design and launch Denial-of-Service and Deceptive attacks to alter the behavior and performance of a 1-DoF robotic arm. The authors in [10] explore the use of a 4-DoF robot manipulator to teach topics on human-robot interaction and leader-follower coordination.

The teaching of high-level control topics such as the coordination of MASs is reported in [11] via the use of a simulation-based urban system. Similarly, the authors of [12] report results on the use of Unmanned Aerial Vehicles (UAVs) in a simulated environment as a tool to motivate undergraduate control engineering education. In [13], the authors present an experimental multi-robot laboratory curriculum using the Khepera IV, while the authors in [14] propose the teaching of the consensus algorithm using the Robotarium at the Georgia Institute of Technology [15].

Replicating the aforementioned experimental MAS laboratory examples can present implementation challenges. For instance, using multi-vehicle systems typically requires a dedicated ample space with restrictions on the number of experiments that can be run simultaneously. Furthermore, conducting experiments with multi-vehicle systems often involves scheduling tasks and student groups, limiting the use of class time for learning. These limitations motivate the development of low-cost, portable laboratory stations that students can easily operate to implement and test control coordination algorithms such as consensus, leader-follower tracking, and formation control protocols.

Networked MAS Testbed

The experimental testbed, illustrated in Figure 1, consists of five 1-DoF rotational robotic links placed one next to the other and mounted on a $60\ \mathrm{cm} \times 12\ \mathrm{cm}$ acrylic frame. Each link has a Hitech HS-475HB servo motor, an Adafruit BNO055 Inertia Measurement Unit (IMU) sensor, and an STM32 Nucleo-L432KC microcontroller (refer to Figure 2). The Hitech HS-475HB Servo motor is the actuator of the robotic link, with a range of rotation of about 180^o. It accepts a Pulse Code Modulation (PCM) signal with a minimum of $500\ \mu\mathrm{s}$ and a maximum of $2500\ \mu\mathrm{s}$ of width. The Adafruit BNO055 IMU sensor is an intelligent 9-Axis absolute orientation sensor that integrates a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a range of $\pm 2000^o/s$, a triaxial geomagnetic sensor, and its own 32-bit ARM Cortex M0+ microcontroller for sensor fusion and processing. It provides angular position information (in this case, the heading of the link in degrees or radians) to the Nucleo microcontroller using the Inter-Integrated Circuit (I2C) communication protocol. An STM32 Nucleo-L432KC microcontroller controls each link. The Nucleo microcontroller can be programmed in C/C++ using the Mbed operating system. Each

Nucleo microcontroller in the testbed is connected to each other via a common CAN bus using an MCP2515-I/P CAN bus adapter. The Nucleo uses Serial Peripheral Interface (SPI) communication protocol to communicate with the CAN bus adapter, which reads and sends messages through the common CAN bus, allowing the implementation of any inter-agent communication graph. Finally, the case housing all components, including the link, is made of plastic using 3D printing. The overall testbed is lightweight, compact, and can be easily transported to any location near a power outlet. It is assumed that the students have a PC or laptop with an open-source terminal emulator program such as Tera Term or RealTerm to communicate with the microcontroller and access to a C/C++ compiler to program the microcontrollers.

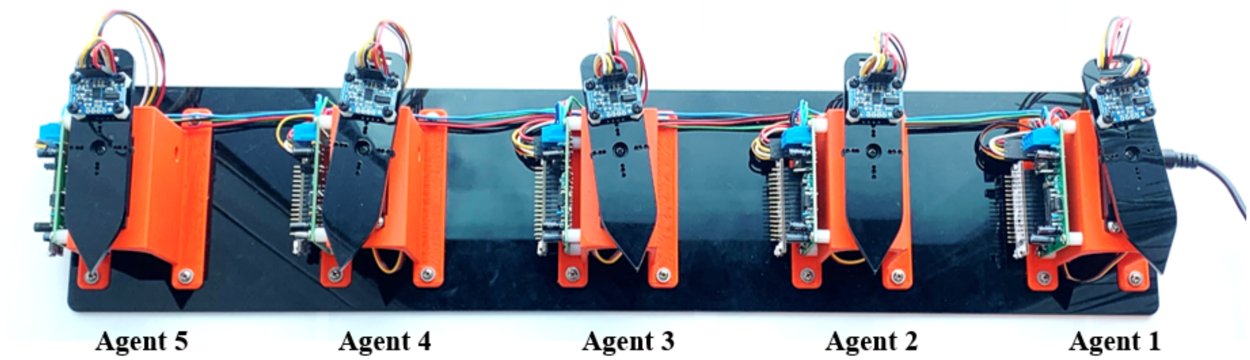


Figure 1: Experimental Testbed

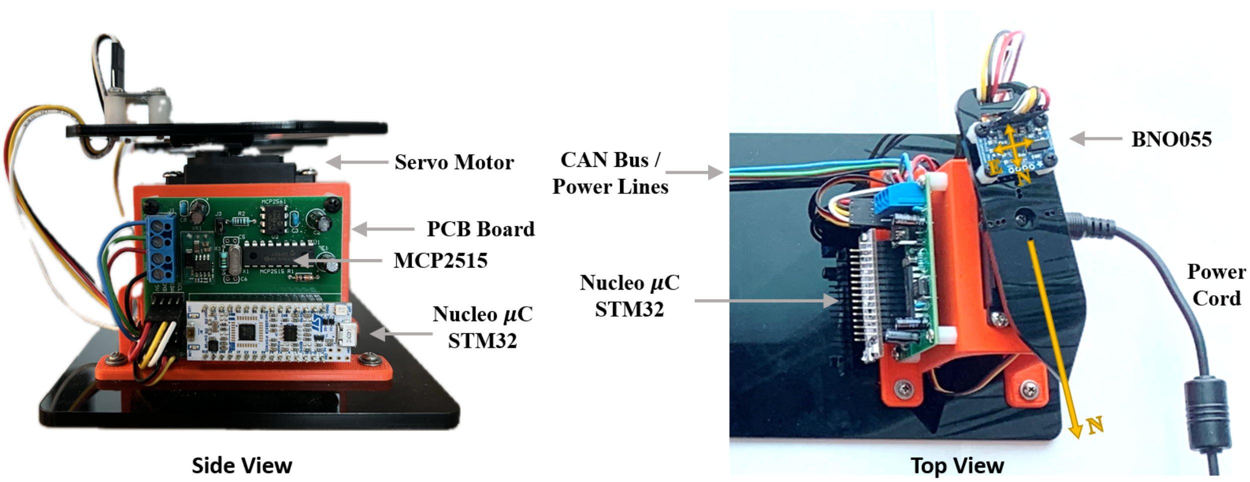


Figure 2: Parts of a Single Rotational Robotic Link

The approximated total cost of the testbed, not including the labor cost to assemble it, is given in Table 2. Several items within the testbed can be substituted in order to reduce cost. For instance, the rotatory links can be mounted in a wood or metal frame, and the microcontroller can be substituted for a cheaper option with I2C and SPI capabilities. Moreover, the IMU sensor can be eliminated: the readings of the IMU sensor can be replaced by estimating the position of the servo motor based on the PCM signal as discussed later under Hands-on Laboratory Activities.

Table 2: Approximated Cost of Parts

Part	**Cost per Unit** (USD)	**Quantity**	**Total Cost** (USD)
HS-475HB Servo motor	20	5	100
BNO055 IMU Sensor	30	5	150
Nucleo-L432KC microcontroller	12	5	60
MCP2515-I/P	3	5	15
PCB Board	10	5	50
12V Power Adapter	10	1	10
Base Frame	20	1	20
Other Parts¶	20	1	20
Total per Unit	–	–	425

¶ Resistor, capacitors, voltage regulators, wiring cables, and screws, among others

Hands-on Laboratory Activities

Students can use the testbed to implement and test the agreement (also known as consensus) protocol [16] under a wide range of communication graphs and communication time delays. Students can also implement other multi-robot coordination algorithms, such as formation control and tracking.

Agreement Protocol

Theoretical Background: Let $\mathcal{G}$ be the communication graph and A the in-degree adjacency matrix of the MAS. The ijth entry of the adjacency matrix, denoted as a_{ij}, is positive if the ith agent receives heading information from the jth agent, $\psi_j(t)$, and zero otherwise. The agreement protocol seeks to drive the headings of all five links to a common value. The protocol implemented by each link's microcontroller is given by

$$\dot{\psi}_i(t) = \sum_{j=1}^{N} a_{ij}(\psi_j(t) - \psi_i(t)) \tag{1}$$

where $\psi_i(t)$ is the heading of the ith link and $N = 5$ is equal to the total number of links. The discrete implementation of (1), with constant sampling interval $T > 0$, is then given by

$$u_i(t_{k+1}) = u_i(t_k) + \sum_{j=1}^{N} a_{ij}(\psi_j(t_k) - \psi_i(t_k)) \tag{2}$$

where $k = \{0, 1, 2, \cdots\}$ are the sample indices, $t_{k+1} - t_k = T$ are the sampling intervals, and $u(t_k)$ is the PCM signal sent from the Nucleo microcontroller to the link's servo motor.

It is well-known that the agents will converge to a common heading if the graph has at least one-rooted-out branching subgraph with a settling time of

$$T_s \leq \frac{4}{\text{real}(\lambda_2)} \tag{3}$$

where λ_2 is the second smallest eigenvalue of the graph's Laplacian matrix [17]. Furthermore, it will converge to the average of the initial headings, denoted as $\psi^{\star}$, if the graph is balanced [17].

Hands-on Activity: Students can be asked to program one or multiple microcontrollers in C/C++ to execute the entire lab exercise or functions of it, such as:

a) Read the IMU sensor.

b) Read and write CAN Messages. Students can be in charge of reading, decoding, and identifying messages through the CAN Bus. Similarly, they can be in charge of encoding and transmitting CAN messages to other agents, depending on their previous C/C++ knowledge.

c) Implement the agreement protocol. Students can code the agreement protocol (1) into the microcontrollers and the adjacency or Laplacian matrix.

d) Compute and send PCM signal to the servo.

e) Program the user interface.

For simplicity, students enrolled in the Control of MASs course at the USNA were provided with most of the code. They were only tasked with programming the adjacency matrix, running the experiment, and collecting and post-analyzing the experimental data.

Students were asked to implement nine different undirected, directed, and weighted communication graphs, all illustrated in Figure 3. Before running an experiment, they were asked to draw theoretical expectations such as convergence value (if any) and settling time (3) using the Laplacian matrix of the communication graph. Once the data was collected, they had to plot it in MATLAB and compare the experimental results with their expectations, compute the error or difference, and draw conclusions. Experimental examples for communication graphs $\mathcal{G}_2$ and $\mathcal{G}_4$ using a sampling interval of $T = 0.1$ s are given in Figure 4.

Alternative without the IMU Sensor: Estimates of the heading relative to the testbed's frame can be obtained using the servo's PCM signal in lieu of the IMU sensor, reducing the cost of the system and the measurements' noise error as seen in Figure 5. The PCM signal and the heading of the link relative to the testbed follows a nearly linear relationship. Each PCM value can be mapped to an approximated heading value as shown in Figure 6.

Multi-Robot Coordination with Communication Time Delays

Implementing the agreement protocol (1) over a shared communication network can bring some challenges, including data losses and time delays. Time delays, in particular, can degrade the overall system's performance and even lead to instability [18]. Therefore, it is important for students to observe and study the effect that communication delays can have on the system's performance.

Two different time-delay cases of the agreement protocol can be studied. The first is when only neighbors' information is subject to a time delay. In this case, the agreement protocol for the ith

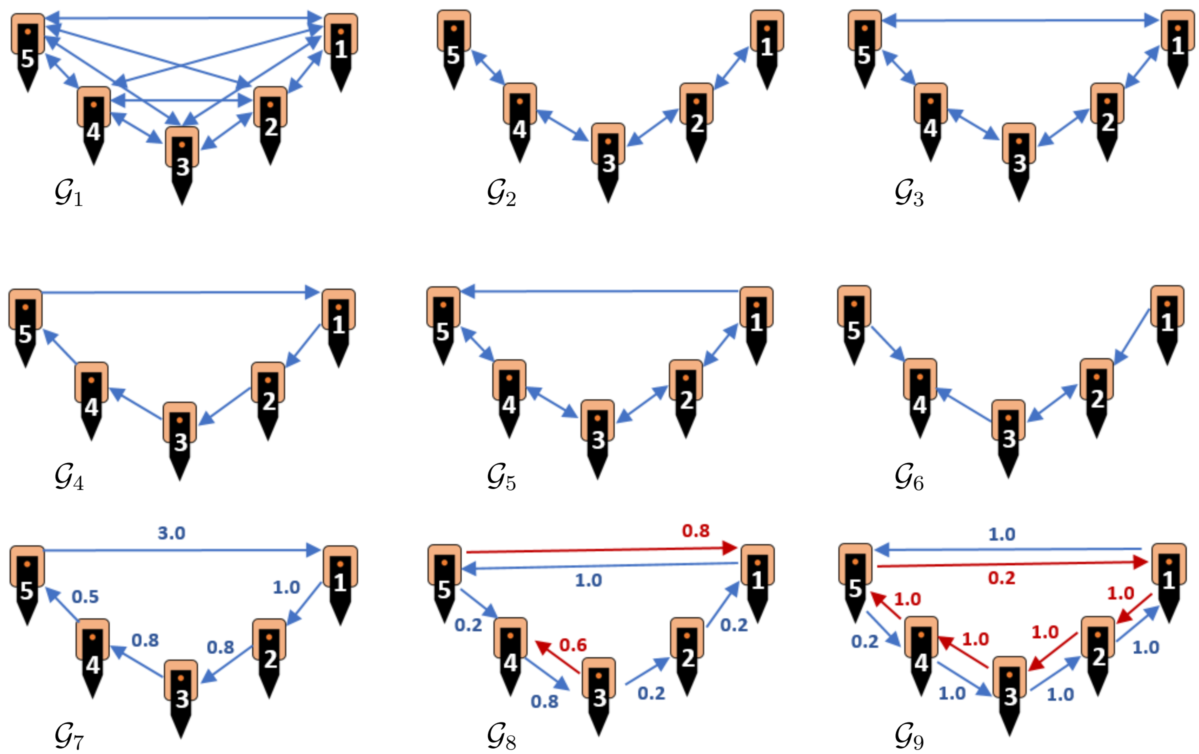


Figure 3: Communication Graphs for Agreement Protocol. Graphs $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ are examples of unweighted, undirected graphs. Graphs $\mathcal{G}_4$, $\mathcal{G}_5$, and $\mathcal{G}_6$ are examples of unweighted, directed graphs. Graphs $\mathcal{G}_7$, $\mathcal{G}_8$, and $\mathcal{G}_9$ are examples of weighted, directed graphs.

agent can be rewritten as

$$\dot{\psi}_i(t) = \sum_{j=1}^{N} a_{ij}(\psi_j(t-\tau) - \psi_i(t)) \tag{4}$$

where $\tau > 0$ denotes the delay. It is well known that the agreement protocol in (4) is stable for any delay; however, the settling time (3) is no longer guaranteed and increases with the delay [16].

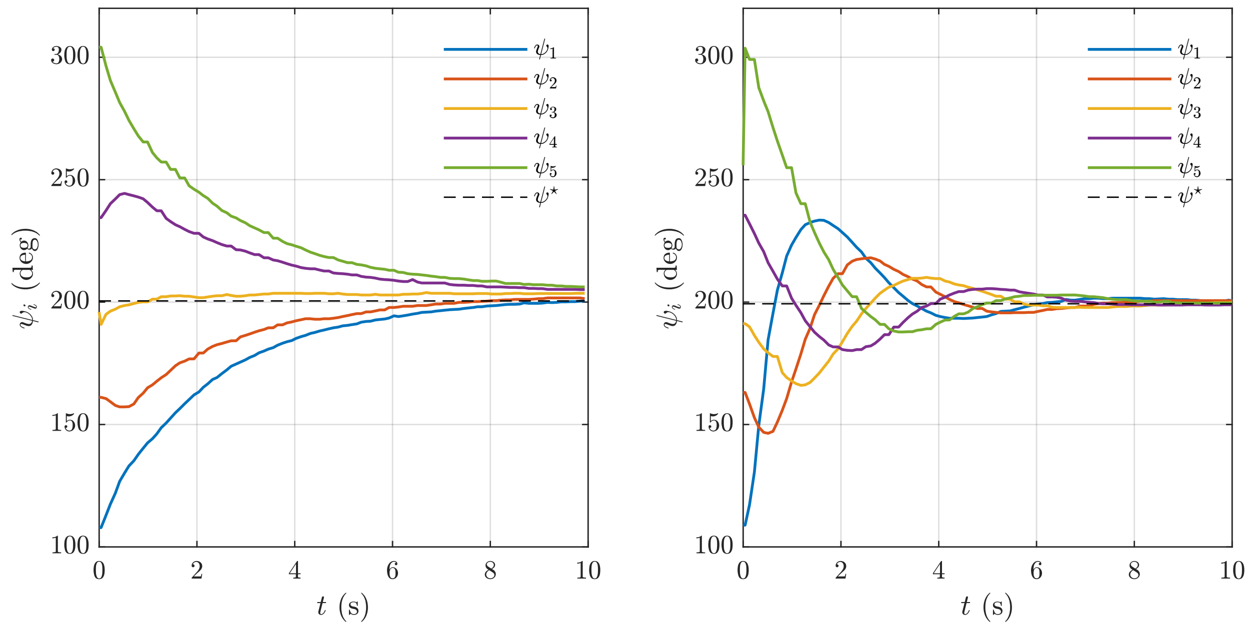
The discrete implementation of (4) is given by

$$u_i(t_{k+1}) = u_i(t_k) + \sum_{j=1}^{N} a_{ij}(\psi_j(t_k - \tau) - \psi_i(t_k)) \tag{5}$$

where $u_i(t_k)$ is the PCM signal and τ is a multiple of the sampling interval T.

The second variation of the agreement protocol represents when information from all agents, including the ith agent, is delayed

$$\dot{\psi}_i(t) = \sum_{j=1}^{N} a_{ij}(\psi_j(t-\tau) - \psi_i(t-\tau)). \tag{6}$$



(a) $\mathcal{G}_2$ (b) $\mathcal{G}_4$

Figure 4: Experimental Results Using the IMU Sensor.

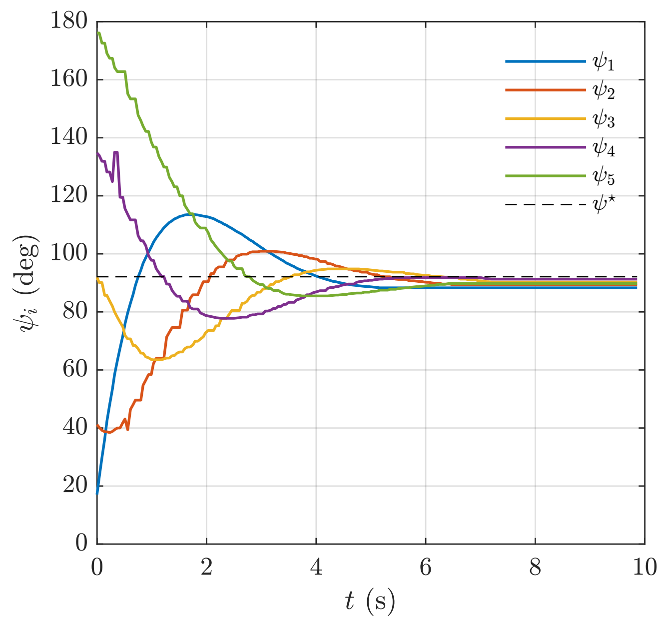


Figure 5: Experimental Results for Communication Graph $\mathcal{G}_4$ Using Heading Estimates from PCM Signal.

In this scenario, the stability of the system is only guaranteed if the delay satisfies the following condition

$$\tau \leq \frac{\pi}{2 \cdot \text{real}(\lambda_{\max})} \tag{7}$$

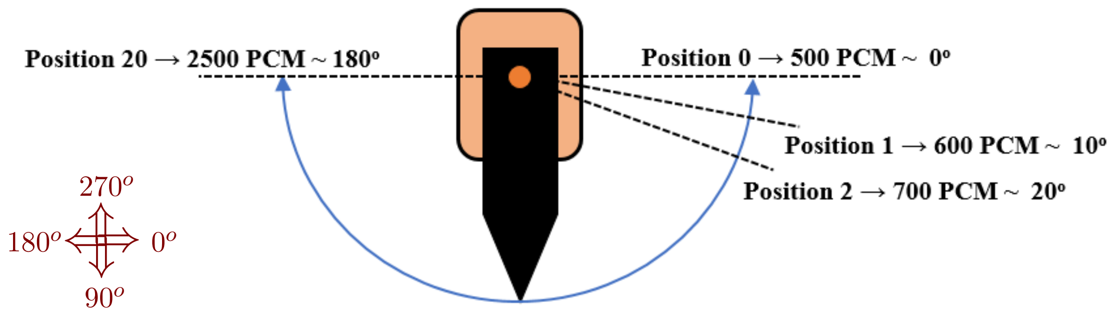


Figure 6: PCM Control Signal and Equivalent Heading.

where $\text{real}(\lambda_{\max}) > 0$ is the real part of the maximum eigenvalue of the communication graph's Laplacian matrix [16].

Similar to (2) and (5), the discrete implementation of (6) is given by

$$u_i(t_{k+1}) = u_i(t_k) + \sum_{j=1}^{N} a_{ij}(\psi_j(t_k - \tau) - \psi_i(t_k - \tau)) \tag{8}$$

where τ is a multiple of the sampling interval T.

Hands-on Activity: In addition to the various tasks presented under the agreement protocol with no delay, students can be asked to:

a) Program the microcontrollers to intentionally incorporate communication delays (e.g., using buffers).

b) Experimentally evaluate the maximum delay that the system can handle without going unstable under the agreement protocol in (6) and compare with the theoretical bound in (7).

c) Study the trade-offs between stability and settling time for different communication graphs.

d) Collect, process, and analyze the data.

Figure 7 showcases an example of experimental results using graph $\mathcal{G}_4$, assuming the time-delay agreement (6) and a uniform time delay of 0.4 s. Note that, compared to Figure 4(b), the settling time is significantly longer.

Other Potential Learning Activities

Leader-Follower Control: Students can be asked to program four of the five agents to follow the heading trajectory of a leading agent. The leader can be programmed with a pre-fixed trajectory or be manually manipulated by the student.

Bilateral Teleoperation: Students can program two or multiple links to execute bilateral control [19], where each link tracks the motion of the other. Students can also evaluate the effects of delays on tracking performance.

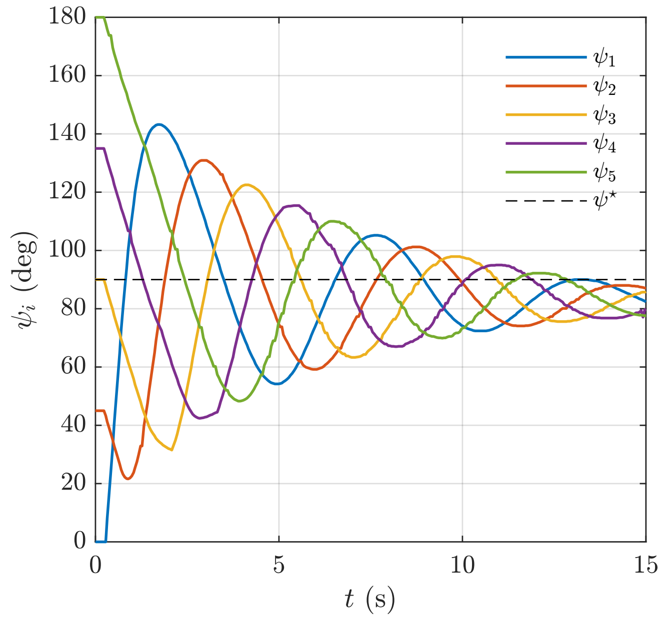


Figure 7: Experimental Results for Communication Graph $\mathcal{G}_4$ under Agreement Protocol (6) and with Time Delay $\tau = 0.4$ s.

Topics on Cyber Security: The effects of intentional malicious attacks on the network can be studied. For instance, students can program a single microcontroller to launch Denial-of-Service and Deceptive Attacks on the system, similar to [9], while trying to achieve consensus.

Learning Outcomes Assessment

The MAS testbed was designed to support two ABET-based learning outcomes within the RCE program at USNA as well as specific learning outcomes of the Control of MAS course. The testbed was used in two different activities during the Academic Year 2025 that are highlighted in Table 1. In total, the students utilized the testbed for at least 6 hours in groups of 2 and 3 students.

The ABET learning outcomes supported by the testbed are:

A1) An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

A2) An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

In addition, the Control of MASs course in which the testbed is introduced, has the following learning outcomes considering the advanced nature of the course:

C1) Understand the control and communication challenges that multi-agent systems face

Table 3: Map of Self-Assessment Questions to Learning Outcomes

Learning Outcomes	Questions/Statements								
	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**
A1	–	–	–	–	–	–	–	√	–
A2	–	–	–	–	–	–	√	–	–
C1	–	–	–	–	√	√	–	–	–
C2	√	√	–	–	–	–	–	–	–
C3	–	–	√	√	–	–	–	–	–

C2) Understand and apply fundamental concepts and tools on graph theory and stability

C3) Design and evaluate consensus control algorithms for multi-agent systems

Once students completed the six-hour exercises and submitted their written report with their findings, a voluntary nine-question form was given to qualitatively assess the testbed's effectiveness and laboratory exercises' effectiveness in achieving the learning objectives. The students were asked to assess their agreement to the following nine statements using a five-point scale (*5=Strongly Agree, 4=Agree, 3=Neither Agree Nor Disagree, 2=Disagree, 1=Strongly Disagree*) in addition to the choice of *Do Not Apply*:

Q1) The testbed and laboratory exercises helped me understand the difference between graph types (e.g., path, cycle, and directed) and different graph properties (such as weakly connected, strongly connected, and disconnected).

Q2) My understanding of graph theory improved after doing the laboratory exercise with the testbed.

Q3) The testbed and laboratory exercises helped me understand and visualize the behavior of the agreement (consensus) protocol under different communication graphs.

Q4) My understanding of the agreement (consensus) protocol improved after doing the laboratory exercise with the testbed.

Q5) Laboratory exercise #4 (Agreement Protocol under Time Delays) helped me understand and visualize the effects of delays on multi-agent systems.

Q6) My understanding of the effects of delays on multi-agent systems improved after doing the laboratory exercise #4 (Agreement Protocol under Time Delays).

Q7) The laboratory exercises helped me improve my teamwork abilities and collaborative skills.

Q8) The laboratory exercises helped me improve my abilities to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

Q9) In general, I liked the use of the testbed for the teaching of graph theory and multi-agent system algorithms.

The map between the statements and learning outcomes is given in Table 3.

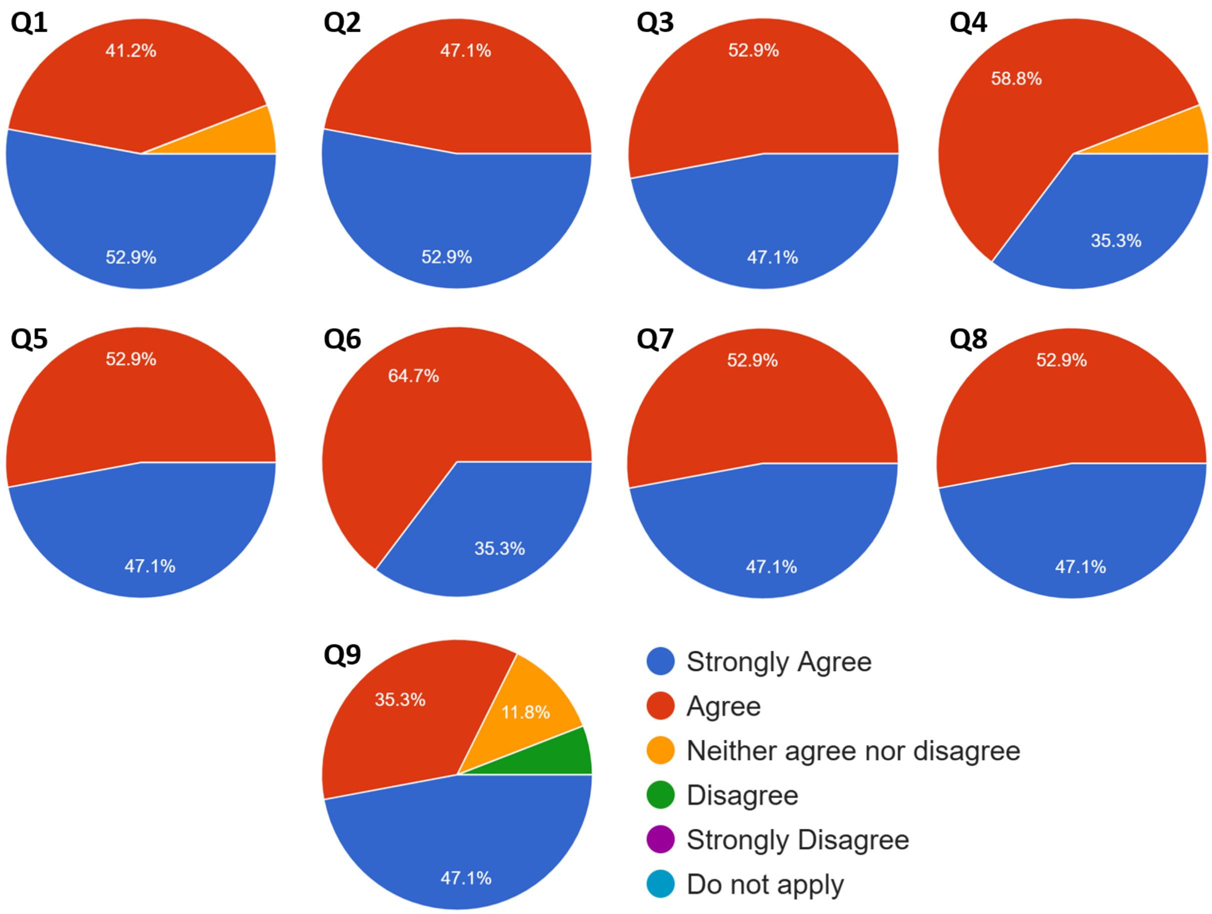


Figure 8: Students Self-Assessment Results.

A total of 17 out of 30 students enrolled in the course voluntarily answered the online form. Their responses were recorded anonymously. The results are summarized in Figure 8. In all statements related to achieving the learning outcomes, 94% or more of the students either agree or strongly agree about the testbed helping them improve their understanding of the course concepts and ABET-related criteria, with the other percentage having no opinion. About 77% agree or strongly agree with linking the use of the testbed as a teaching method for graph theory concepts, with only less than 6% disagreeing.

Conclusions

This paper presented a low-cost, portable MAS testbed for the teaching of graph theory and networked control of multiple agents. Students can implement and test multi-robotic algorithms such as consensus and tracking. Furthermore, they can study the effect of delays in the communication network with the potential of exploring other network-based vulnerabilities such as data losses, network congestion, and cyber attacks. Results of a self-assessment questionnaire among students who have interacted with the testbed provided evidence of the benefits of incorporating such a testbed into an undergraduate control engineering curriculum.

Acknowledgments

The authors would like to thank the Technical Support of the Weapons, Robotics, and Control Engineering Department at USNA, in particular, to Joseph Bradshaw, Daniel Rhodes, John Seargeant, and Zac Cohen for their support in developing and building the testbed.

References

[1] D. Hristu-Varsakelis and W. Levine, "An undergraduate laboratory for networked digital control systems," *IEEE Control Systems Magazine*, vol. 25, no. 1, pp. 60–62, Feb. 2005.

[2] M. Törngren, M. E. Grimheden, J. Gustafsson, and W. Birk, "Strategies and considerations in shaping cyber-physical systems education," *ACM SIGBED Review*, vol. 14, no. 1, pp. 53–60, Jan. 2017.

[3] C. Berry, M. Gennert, and R. Reck, "Practical skills for students in mechatronics and robotics education," in *Proc. ASEE Annual Conference Exposition*, 2020.

[4] J. M. Esposito, "The state of robotics education: Proposed goals for positively transforming robotics education at postsecondary institutions," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 157–164, 2017.

[5] S. Jung, "Experiences in developing an experimental robotics course program for undergraduate education," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 129–136, 2012.

[6] S. Gröber, M. Vetter, B. Eckert, and H.-J. Jodl, "Experimenting from a distance—remotely controlled laboratory (rcl)," *European Journal of Physics*, vol. 28, no. 3, p. S127, 2007.

[7] M. Garduño-Aparicio, J. Rodríguez-Reséndiz, G. Macias-Bobadilla, and S. Thenozhi, "A multidisciplinary industrial robot approach for teaching mechatronics-related courses," *IEEE Trans. Educ.*, vol. 61, no. 1, pp. 55–62, Feb 2018.

[8] E. J. Rodríguez-Seda, P. J. Frontera, and J. Bradshaw, "A networked cyber-physical system testbed for undergraduate education," in *Proc. Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 3007–3012.

[9] P. J. Frontera and E. J. Rodríguez-Seda, "Network attacks on cyber–physical systems project-based learning activity," *IEEE Transactions on Education*, vol. 64, no. 2, pp. 110–116, 2020.

[10] V. Petit-Magat, H. Lhachemi, C. Stoica, A. Thakker, and M. Da Silva, "Enthusing engineering students over multi-agent systems control via human-robot interaction," in *Proc. IEEE International Conference on System Theory, Control and Computing*, 2023, pp. 24–29.

[11] F. Soygazi, "Multi-agent systems and machine learning for wind turbine power prediction from an educational perspective," *Sustainability*, vol. 15, no. 23, p. 16291, 2023.

[12] C. S. Maniu, C. Vlad, T. Chevet, S. Bertrand, A. Venturino, G. Rousseau, and S. Olaru, "Control systems engineering made easy: Motivating students through experimentation on uavs," in *Proc. IFAC World Congress*, 2020.

[13] G. Farias, E. Fabregas, E. Peralta, H. Vargas, S. Dormido-Canto, and S. Dormido, "Development of an easy-to-use multi-agent platform for teaching mobile robotics," *IEEE Access*, vol. 7, pp. 55 885–55 897, 2019.

[14] S. Bertrand, C. S. Maniu, and C. Vlad, "Teaching by practice the basis of consensus for multi-agent systems," in *Proc. IFAC Workshop on Aerospace Control Education*, 2021, pp. 20–25.

[15] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[16] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[17] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010, vol. 33.

[18] S.-I. Niculescu, *Delay effects on stability: a robust control approach*. Springer, 2003, vol. 269.

[19] E. J. Rodríguez-Seda, J. J. Troy, C. A. Erignac, P. Murray, D. M. Stipanovic, and M. W. Spong, "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 984–992, 2009.