Borge, J., Sauar Wad, M., Økter, M., Ottestad, M., Rudolfsen, M.H. & Zhou, J. (2022). Design and Implementation of Mechatronics Home Lab for Undergraduate Mechatronics Teaching. 2022 IEEE 17th International Conference on Control & Automation, 291-296. https://doi.org/10.1109/ICCA54724.2022.9831876

Published in: IEEE International Conference on Control & Automation

DOI: https://doi.org/10.1109/ICCA54724.2022.9831876

AURA: https://hdl.handle.net/11250/3029381

Design and Implementation of Mechatronics Home Lab for Undergraduate Mechatronics Teaching

Jorgen Borge, Martin Sauar Wad, Martin Økter, Morten Ottestad, Morten Hallquist Rudolfsen, Jing Zhou

Abstract— **The field of mechatronics is a multidisciplinary field of engineering, where the combination of physical components and theory from several engineering fields is applied to build complex machines. Mechatronics education is an active learning process through practical laboratory exercises and problem-based learning. This paper presents the design and implementation of a mechatronics home lab to support undergraduate mechatronics teaching. The purpose is to support theoretical teaching in mechatronics with a low-cost, 3D-printable platform where the students can experiment and practice instrumentation and control theory with a practical problem-based approach. Five projects were introduced for experiment implementation of the developed home lab. Throughout these experiments, it is intended to facilitate the understanding of theories and concepts in mechatronics, and enhance the ability to design and implementation of experiments, the collection and analysis of data, and the conducting of simulation in MATLAB.**

Keywords: Mechatronics education, laboratory, equipment, undergraduate, active learning

I. INTRODUCTION

The field of mechatronics is a practical study as much as a theoretical study, where the combination of physical components and theory from several engineering fields are applied to build complex machines. A pivotal pillar to mechatronics education is an active learning process through practical laboratory exercises and problem based learning, [1], [2], [3], [4], [5], [6], [7]. Traditionally, a number of these exercises involve experiments with expensive stationary equipment, scheduled time, fixed location, etc., as in [1].

In connection with pandemics caused by COVID-19 or similar incidents that prevent students from being physically present at universities, teaching and many activities are carried out digitally without laboratories access. In this vulnerable situation, the students must maintain a concrete relationship with the subject. In this situation, home lab will physically link the students to the theory in the subject.

The purpose of the Mechatronics Home Lab is to accommodate the theoretical curriculum in mechatronics with a low-cost, 3D-printable platform where the students can experiment and practice instrumentation and control theory [8] with a practical problem-based approach. The platform involves different experimental setups with a moderate increase in difficulty, from a simple temperature control of a heating element to a hardware-in-the-loop (HIL) simulation and control of a drone position in 3D, by controlling the pitch, roll, and thrust of a stationary drone. Throughout these experiments, the students follow David Kolb's four point learning cycle [9] described in figure 1.

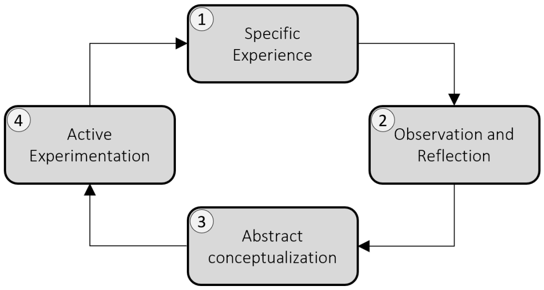

Fig. 1: Kolb's Learning Cycle

The learning cycle starts with a supervised experiment where the students gain some specific experience. Through the experiment, causes and effects are observed and reflected upon. Based on the observations and previously lectured theory, the students form concepts and abstract models. Taking the concepts and models into active experimentation, the knowledge of the experiment expands, leading to an increase in the complexity of the problem. By increasing the complexity of the problem, new specific experience is obtained, and the cycle continues. The cycle is repeated multiple times with each experimental setup in the Mechatronics Home Lab. Thus, each experimental setup itself build on knowledge from previous experimental setups.

II. COMPONENTS AND OVERVIEW

The Mechatronics Home Lab is based on a fundamental setup including an Arduino-micro-controller, a breadboard, a motor-controller (L298N) and a power source. In addition to these components, a wide range of sensors and actuators can be added to make the setup able to support a number of different experiments. The add-ons for this specific system are a load-cell, op amp, IMU, current-sensor and DC-motors.

To organize these components orderly, a plate was designed based on the intention of being 3D-printed. An advantage with this design is the components can be appropriately placed and secured in place. All components are secured by the use of M3 machine bolts. This prevents the parts from moving during experiments, transport and storage,

---

Authors are with the Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway. Email: martino@student.uia.no; morten.rudolfsen@uia.no; morten.ottestad@uia.no; jing.zhou@uia.no

as well as minimizing the relative movements. A major reason for error is situated in the wires and connections between the components, due to the size of the wires and the use of Dupont connections. The factor of minimizing the relative movement is therefore crucial in creating a setup with minimal and predictable noise in the system. There are built-in pathways, in terms of holes through the dividers beneath the plate, for all wires necessary. Which, in addition to clean up the surface, allows for all signal-wires to be separated from the power-connections. By securing the wires in a fixed location, thus separating power from signal lines, both capacity and flux noise are kept at an acceptable level. This makes the setup suitable for experiments over a longer time duration, making that the conditions for reliable gathering of data unchanged if stored or transported.

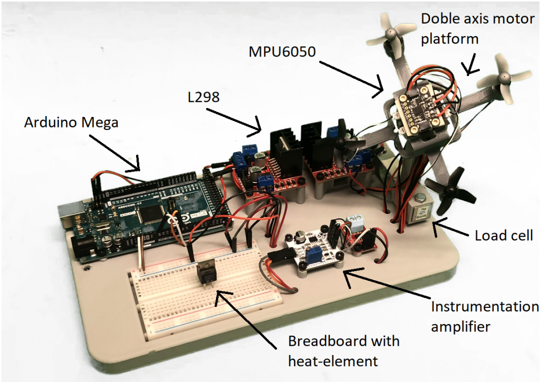

Fig. 2: Complete standard setup of the standard equipment

All mechanical parts of the setup are designed to be 3D-printed on a standard quality printer, which is easily accessible and not too expensive nowadays. By taking advantage of the resent revolution in cheap and precise 3D-printing, it is possible to create a simple design that everyone with access to a 3D-printer can manufacture their own parts. Usually based on open source STL-files. It is also possible for universities and other educational institutions to print and distribute the parts in a larger quantity.

The plate, which the components are attached to, is the only 3D-printed part absolutely necessary. In addition, there are several other available smaller components designed for 3D-printing, used to attach other platforms and sensors experiment-specific to the setup if needed. One of the connection points are designed for attaching a load-cell to the plate. This connection also serves as an optional attaching point for other tools, for easy implementation to the setup. If only the load cell signals are to be investigated, a weight cup may be mounted for containment of weight during testing.

An experiment-tool, which is designed for the plate, is a platform with four possible connection points for DC motors, in a plus-configuration. It is mounted to a tower through a double axis system, with bearings connected to the tower. This configuration may serve as a platform for simulating a flight-controller for a drone. By attaching the tower to a load-cell mounted to the plate, all 3 axis of freedom regarding drone-flight characteristics may be simulated. Further, a basic fight-controller based on different types of regulating systems, may be tested, tuned and investigated. Later, the resulting physical system may be tested and tuned in a safe environment.

There is possible to use a wide range of microcontrollers for this outline. In the initial setup, an Arduino Mega 2560 is used because of its characteristics with many output pins and slightly higher clock rate than the Arduino Uno. It is however integrated screw holes to hold a Uno for easy implementation, if less clock-rate are demanded and a cheaper cost wanted. There is also possible to use higher clock rate versions such as the Arduino Due or even other manufactures as Raspberry Pi or ESP32 for better performance. Creating a more advanced system may be needed for projects that deal with, for instance, a magnetic levitation control-system.

The speed controller mainly used is the L298N, which provide enough current for the motors to be driven at high performance. There are implemented two setup points on the plate, sized for mounting a L298N each, with a power plug easily attached beneath. This is a well tested, but somewhat old motor-controller and has a disadvantage of building up heat. This may affect the operation time at higher speeds, but different heat sinking operations may be implemented for enhanced performance. The L298N is easily available at a low cost, which makes the setup easily available and easy to use.

TABLE I: Basic components

Amount	Part Type	Description
1	Arduino Mega	Controller unit to read and write signals, also computing PID-controller.
1	L298N	Motor controller used to control current in a circuit.
1	Breadboard	Board to organize the wiring.
1	12V power supply	Power supply to the system.
2	Female to male wire	2.54 mm female to male jump wire.
1	Jump wire sett	Jump wire sett to connect components.

In table I the basic components for the design are listed, while table II describes the components needed for conducting the experiments described in section III. In addition, the setup can be fitted with a large range of different components, and are highly adjustable. Which gives opportunities to further develop the described experiments, and making room for more advanced control. The plate may also be used as a platform for developing new control system experiments by adding components by choice.

In figure 3 the communication lines for data acquisition are illustrated. A computer is linked to a microcontroller, either for direct programming or uploading of a program based on the programming platform chosen. Further, the microcontroller is connected to a motor controller with jump wires, communicating by PWM. The motor controller adjusts

TABLE II: Recommended components for the project

Amount	Part Type	Description
1	LM35	In addition with 25Ω resistor used as a heater.
1	ACS723	Current sensor to measure current in a circuit.
1	Instrumentation Amplifier	Instrumentation Precision Amplifier with 500 times amplifier.
1	Load cell	Load cell measuring up to 1 kg.
4	Brush motor	8.5x20 mm 53,500 RPM Brush motor.
4	Propeller	4 blade 31 mm propeller.
1	Silicon wire	4 meters 30AWG silicon wire.
1	MPU6050	9 axis IMU.
4	Ball bearings	623Z ball bearing, 10x4x3 mm.

the current transmitted to the motors. An implemented IMU reports back to the microcontroller using I2C and on to the computer.The microcontroller is reading a 5 volt analog signal from an instrumentation amplifier in extension of a load cell. If needed, the microcontroller would be able to communicate with almost any additional sensors added.

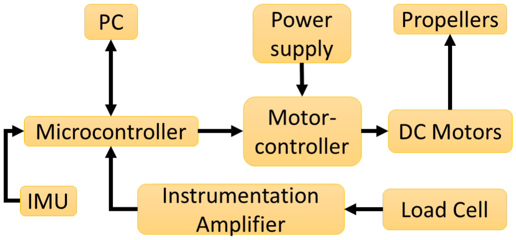

Fig. 3: Communication lines for data acquisition for the complete system

III. Proposed Laboratories and Projects

There are several possibilities to preform different types of experiments regarding this project. However, the design is optimized around the implementation of five specific subprojects. Although there is room for adjustments, both in the described projects and around the possibility for adding new ones. The original five subprojects will briefly be explained in this section, using the basis for the setup intended for the Mechatronics Home Lab equipment.

These experiments are originally accomplished by using MATLAB Simulink programming tool with the Arduino tool box installed. When running the program, the "Monitor & Tune" method are used. Meaning, the program is compiled in such a way that only the IO-pins on the Arduino are used and sent by serial communication to the computer. All calculations are made by the computer. This enables a high computing power for more complicated systems. The downside is the time used for communication, which may make achieving a real time system difficult for more advanced systems. In such cases the microcontroller may be upgraded, allowing for onboard computing, eliminating the need for communication.

A. First Order Temperature Control

The purpose with the initial subproject is to understand and take in use a temperature regulating system, along with how this can be controlled by using hysteresis-regulation and PID-regulation. One may start with controlling the application to a resistor through a motor-controller operated by an Arduino. The resistor is glued together with a temperature-sensor, the current going through the resistor entails an increase in temperature which the temperature-sensor will read and then provide the raw-data signal. This signal needs to be modulated with some mathematical functions for the output to be read in degrees instead of bit. As a result of the prementioned aspects, a system that can control the temperature, through the motor controller L298N, will be created.

One of the simplest ways to control the heat-element is to make a hysteresis-regulator. In principle, this works as a relay which turns the current off when the value is above the desired top point and back on when the value is below the desired bottom point. The top-point and bottom-point are calculated as follows: $H_{top} = SP + \frac{H}{2}$, $H_{bot} = SP - \frac{H}{2}$, in the equations SP stands for the set-point and H is the given hysteresis.

Hysteresis-regulation is a good fit for controlling temperature as a result of the system being slow with predictable values. However, it is possible to use a PID-controller to regulate the system. A PID controller are suitable for fast systems which requires a precise and responsive regulation. The PID controller will work fine in process of temperature-control, nevertheless will its real potential not be shown within regulation.

PID stands for Proportional, Integrator and Derivative, they are representing respectively "present tense", the "past" and the "future". An interaction between these three are calculating an optimal application gain. By creating a simple feedback-loop the error will be calculated based on the set-point and the actual value, further will the PID-controller integrate and derivative the error. Then the calculations of the error will be multiplied with the corresponding gains. The sum of these three values, which is the gained PID, represents the gain. This loop continues until the set-point is reached and continues to sustain the temperature by continuously correct for environmental impact.

The system can be modeled as shown in Figure 4 with only one feedback loop. In the physical system, this loop consist of the heat element which makes the system warmer and a temperature sensor measuring the system temperature. The system have one input coming from the desired set point. Together with the measured temperature, the error is calculated. This value goes further to the PID block and a steering signal is calculated, and further provided for the

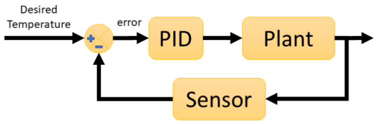

Fig. 4: Model of the basic regulation system

plant. The system is described as a first order system, shown in equation (1).

$$T(s) = \frac{K}{\tau_s s + 1} \tag{1}$$

$$K = \frac{T_f - T_i}{P} \tag{2}$$

where K is the gain, τ_s is the time constant, T_i is the initial temperature, T_f is the final temperature, and P is the applied power.

By first exposing the system for a constant of on cycles and tracking the temperature, key elements as the gain and time-constant may be calculated. The time constant is found at 63 % of the final temperature. When the system is emitting the same amount of heat as produced, the steady state temperature can be used in equation (2), where P is the applied power, and the system gain is calculated [10]. The found model for the system, based on these equations, may then be exposed for a similar step-input and compared to the real results. For safety of the system, a duty cycle of 25 % is used, which have to be taken care of in the equations additionally.

B. Load Cell Calibration

The purpose with the load cell subproject is to establish a mathematical function of the load cell. The load cell will be used for measuring thrust, which opens the opportunity for simulating the altitude in subproject III-E. The measuring will be done by using the Arduino combined with an instrumentation amplifier, also known as an op amp, which is connected to the load cell. In this way, the read signals can be converted to force by modulating the raw data. A good representation of the signals could be a graph. A graph will make the visualization of the calculation much easier to understand.

Further one need to balance the bridge circuit at zero load, this is done to compensate for the tare weight of the weight bowl, as well as any biases which may cause errors in the signal. While adjust the zero point, it is important to consider that the instrumentation amplifier only writes positive signals. If the signal is below zero, the output will still show zero – although the proper signal is negative. This will lead to an error of an unknown size. When this is accounted for, a solution is to make the signal one bit above, such as the error possesses a known size.

The mathematical relation should be based on gram as the unit, however the output signal from the load cell is in bit – and extends from 0 to 1023. As a consequence, the signal need to be converted to gram. To get empirically substantiated data for the relation, one can take several known masses and read the bit signal for each single measurement. These data will further make a table of points which shows the correlation between bit and gram. A graph-program should then be used to make a regression line based on the correlation data from the table. The load is a proportional function, which leads into a regression line represented by $y = Ax$. This equation will then be implemented to the readings from the op amp, which converts the output to gram by multiplying the signal with A. For reliable values the validity of the load cell should be confirmed. This is achieved by adding known masses in ascending order and then remove the masses in an opposite order. The values should then be the same both in ascending and descending order around the apex, if so the validity is confirmed.

C. Propeller Thrust Control

During this subproject, the Arduino shall be used for measuring the bridge-circuit with different thrust-loads, simultaneously measuring the current-consumption. In the previous experiment the load cell measured load downwards, since the thrust pulls the load cell upwards the signal-cables from the load cell to the instrumentation amplifier should be opposite to the previous experiment. The tare weight will now be different, hence the different masses between the weight-bowl and the drone assembly, therefore do one need to preform a new zero point adjustment. To get the motors running, the program has to generate PWM-signals. Later on, some trust-values are needed to calculate parameters for the propeller. A series of different PWM-values should be run while noting the trust-values along with the PWM-signal in a table. These calculations will give several relations within the system including current, lifting power, RPM, and torque based on the specifications to the propeller.

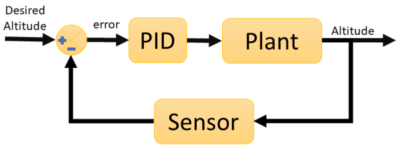

Fig. 5: Model of the thrust control system

The system may be modeled as shown in figure 5 and have a second order figuration. The order is due to the use of an inner loop adjusting the current based on an outer loop measuring the force that the system is providing. Knowing the weight of the system, the height may be simulated. The set point is set as the height the system is expected to provide, together with the measured force from the load cell the necessary corrections are calculated.

In this part, the control objective is to stabilize the height. Using Newton–Euler method and linearization, the transfer function between the height $z(t)$ and applied torque $u(t)$ is given by,

$$G(s) = \frac{Z(s)}{U(s)} = \frac{1}{ms^2} \tag{3}$$

where m is the mass of the drone assembly.
Key values for the motor current to force relation may be found from the measured data points. Taking advantage of the found equations for the system together with the calculated parameters, a simulated system based on the equation can be made. This system may be tested against the same input and compared to the real system.

D. Drone Pitch Control

During the subproject III-D Pitch control, the intention is to get control over a stationary drone using the raw-data from the IMU, which is short for Inertial Measurement Unit. These outputs give the acceleration and angular velocity to the respectively three axes. A composition between them is essential to extract the accurate pitch-angle. A complementary filter is a way of achieving this. The filter fuses the strengths of different sensors, in this case a gyroscope and an accelerometer. More specifically, is this multisensory data fusion and can provide a higher bandwidth, better accuracy and prevent drift error over time. The complementary filter is represented by equation (4). [11]

$$\theta_n = \alpha(\theta_{n-1} + \dot{\theta}_g dt) + (1 - \alpha)\theta_a \tag{4}$$

The angle estimate is given by θ_n, and θ_{n-1} is the previous iteration angle estimate. The filter constant, α, is a number between 0 and 1 which determines the weight of which sensor the filter should prioritize and the response of the filter. The angle estimated from the accelerometer is represented by θ_a, and $\dot{\theta}_g$ is the angular velocity given by the gyroscope. Figure 6 describes the implementation of the complementary filter in Simulink as a sub-system used in the control system for this subproject.

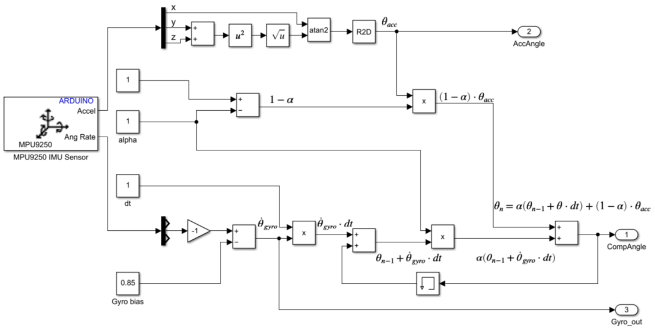

Fig. 6: Simulink sub-system: Complementary Filter

An example of the Simulink code used for this subproject is shown in figure 7, which is based on the model described in figure 8. The figure shows how the plant involves multiple blocks in Simulink, from instance estimating the angle described in figure 6, or sending duty-cycle signal to the PWM-blocks for controlling the motors.

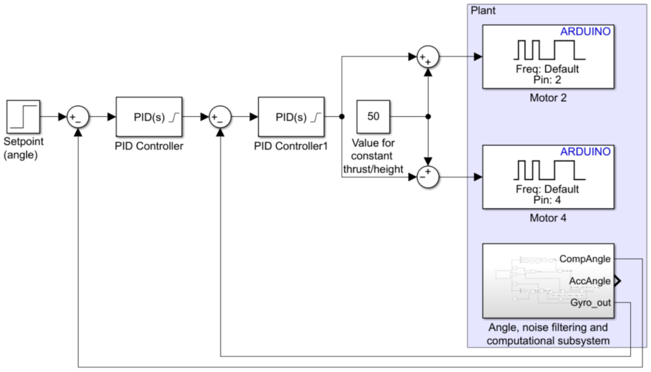

Fig. 7: Simulink block-program for controlling pitch-angle

The purpose with this subproject is to acquire accurate control over the pitch-angle, which further will be used to simulate a stationary drone within a plane. By integrating the angle, one will achieve the horizontal velocity and position through the use of Newton's laws of motion. By the use of cascade control can one determine a set point to the position of the drone, and further generate a set point for the velocity loop.

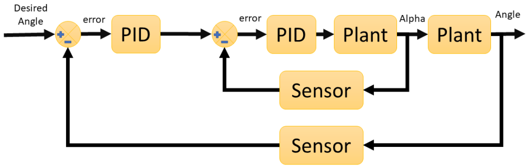

Fig. 8: Model of pitch or roll control

In Figure 8 a block diagram of the system is shown. The system can be modeled as a second order system using angle values from the accelerometer together with the angle velocity values from the gyroscope. Both values provided by the IMU unit. A desired angle is provided to the system as a constant and used for the set point of the angle velocity. This is combined with the measurement of the current angle velocity, which goes through a PID controller, resulting in a PWM steering signal for the motor controller.
The transfer functions for pitch and yaw angles are derived using Euler-Lagrange equations and linearization as in [12], given by

$$G_{pitch}(s) = \frac{\Theta(s)}{\tau_p(s)} = \frac{1}{J_p s^2 + D_p s + K_{sp}} \tag{5}$$

$$G_{roll}(s) = \frac{\Psi(s)}{\tau_r(s)} = \frac{1}{J_r s^2 + D_r s} \tag{6}$$

where $\Theta(t)$ and $\Psi(t)$ are pitch and roll angles in radians, τ_p and τ_r are torques acting on pitch and raw axes, J_p and J_r are the total moments of inertia of the pitch and roll respectively, D_p and D_r are the damping constants for the rotation along pitch axis and the raw axis, K_{sp} is stiffness along pitch axis. [12].

If the moment of inertia in the system are known together with the damping ratio, equations (5)-(6) can be used for modelling the system. The damping ratio may be calculated from empirical testing and the moment of inertia calculated from the top ring weight distribution. The terms τ_p and τ_r represent the torque emitted by the motors at each axis of the rig. The difference between the force emitted by the motors, multiplied with the distance from the center of the rig to the center of a motor, results in the torque which the rig is exposed to. This system may then be used for calculating optimized PID values, which then can be compared to the empirical found values of the real system, creating a connection between empirical and theoretical found values.

E. Stationary Drone Control, 8-DOF Simulation

The drone assembly shall now be used to simulate an autonomous drone, which can move to any set waypoint in 3D-space. This experiment is quite similar to section III-D, however simulation of the roll-angle and the altitude will now be included. The modulation of the roll-angle is done the same way as the pitch-angle, which was explained in the previous subsection. The altitude will be simulated based on section III-C. Since the drone assembly stays fixed during the simulations, it is impossible to use GPS for traction. Therefore, a cascade control is used to make the drone fly the decided distances with higher precision.

At this stage, all degrees of freedom for a real life drone are simulated and controlled, with an exception for the rotation around the upwards axis (”yaw”). However, it is not necessary for basic maneuvering. The yaw axis also represents a more complex steering, as it in real life relay on a magnetometer for third view correction, introducing a higher amount of noise. The relative movement from the staring point may then be plotted, or with advanced software modeled in a simulated environment.

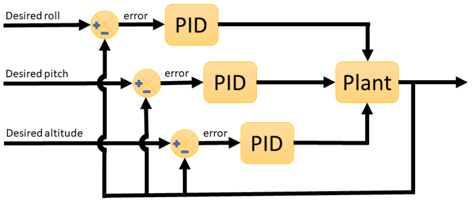

Fig. 9: Model of pitch, roll, and altitude control

The system is at this stage a combination of several models, which are previously calculated and modeled. The total system controlling the flight of the drone assembly may be seen as a multiple-input, multiple-output system (MIMO). A block diagram is presented in Figure 9. For the input three desired set point, namely for pitch, roll, and height are sent to individual PID blocks and provided as correction signals for the plant. The control structure is a decentralized PID control for MIMO system. The system can be further represented in a state-space model and using a state feedback control [13], this is relevant for a master course.

The equations are the same for pitch and roll axis as previously discussed in the pitch control section III-D, and is used as a correctional speed for the motors. For the baseline of the motor power, the thrust control equations are used as described in the thrust control chapter III-C. Together, this system will describe the behavior of a simulated drone, in eight degrees of freedom.

IV. CONCLUSION

This paper describes the procedure to build a mechatronics home lab. With this, the teaching with experiments can be maintained continuity in a virtual format. Throughout the home lab, the students learn mechatronics with the mindset of “sense, think, act” where sensors, actuators and software are used to solve practical problems within the instrumentation, measurement and control theory. By overcoming practical challenges and issues with hardware, the students develop skills and get familiar with systematic methods of troubleshooting they would never experience in simulations, which improve the learning outcome of different subjects involved.

REFERENCES

[1] Z. Lei, H. Zhou, W. Hu, and G. P. Liu, “Teaching and comprehensive learning with remote laboratories and matlab for an undergraduate system identification course,” *IEEE Transactions on Education*, early access, 2021.

[2] C. C. Dosoftei and A. E. Cojocaru, “Implementation of a virtual control lab to support teaching in engineering control,” in *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*, 2020, pp. 699–703.

[3] O. Onel, K. Evans, P. Green, P. Kyabaggu, J. Medland, R. Smalley, B. Lennox, and O. Marjanovic, “Development of a teaching laboratory for undergraduate control systems teaching,” in *UKACC International Conference on Control*, 2010, pp. 1–6.

[4] P. M. Oliveira and J. D. Hedengren, “An apmonitor temperature lab pid control experiment for undergraduate students,” in *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 790–797.

[5] S. Khan, M. H. Jaffery, A. Hanif, and M. R. Asif, “Teaching tool for a control systems laboratory using a quadrotor as a plant in matlab,” *IEEE Transactions on Education*, vol. 60, no. 4, pp. 249–256, 2017.

[6] R. M. Reck, “Common learning objectives for undergraduate control systems laboratories,” *IEEE Transactions on Education*, vol. 60, no. 4, pp. 257–264, 2017.

[7] C. Chandrasekara and A. Davari, “Control experimentation for undergraduate students,” in *Proceedings of the American Control Conference*, vol. 7, 2005, pp. 5156–5161.

[8] N. S. Nise, *Control Systems Engineering*. 6th edition. Pomona: John Wiley & Sons: Wiley, 2017.

[9] D. A. Kolb, “Experience as the source of learning and development,” *Upper Sadle River: Prentice Hall*, 1984.

[10] J. Borge, M. S. Wad, and M. Økter, *MAS240 - Mechatronics Technical Report*. Grimstad, Norway: University of Agder, May 2021.

[11] P. Gui, L. Tang, and S. Mukhopadhyay, “Mems based imu for tilting measurement: Comparison of complementary and kalman filter based data fusion,” in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 2004–2009.

[12] S. M. Schlanbusch, *Adaptive Backstepping Control of Quanser 2DOF Helicopter : Theory and Experiments*. Master’s thesis in Mechatronics, University of Agder, Norway, 2019.

[13] S. M. Schlanbusch, J. Zhou, and R. Schlanbusch, “Adaptive attitude control of a rigid body with input and output quantization,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 8296–8305, 2022.