

2-7-2023

## Utilization of Simulation Software to Enhance the Learning Experience for Students at the Worldwide Campus

Kimberly Luthi

*Embry-Riddle Aeronautical University, kimberly.luthi@erau.edu*

Christian Janke

*Embry-Riddle Aeronautical University, jankec@erau.edu*

Yuetong Lin

*Embry-Riddle Aeronautical University, liny4@erau.edu*

Follow this and additional works at: <https://commons.erau.edu/publication>



Part of the [Educational Technology Commons](#), [Online and Distance Education Commons](#), and the [Science and Mathematics Education Commons](#)

---

### Scholarly Commons Citation

Luthi, K., Janke, C., & Lin, Y. (2023). Utilization of Simulation Software to Enhance the Learning Experience for Students at the Worldwide Campus. , (). Retrieved from <https://commons.erau.edu/publication/2010>

This Presentation without Video is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Publications by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

# 2023 Academic Innovation Virtual Conference

▶ Roundtable Discussion: Utilization of simulation software to enhance the learning experience for students at the Worldwide Campus

▶ **Presenters:**

Christian Janke, Assistant Professor of the Practice, College of Aviation

Dr. Kimberly Luthi, Assistant Professor, College of Aviation

Dr. Yuetong Lin, Associate Professor, College of Aviation

# Introduction and Disclaimer

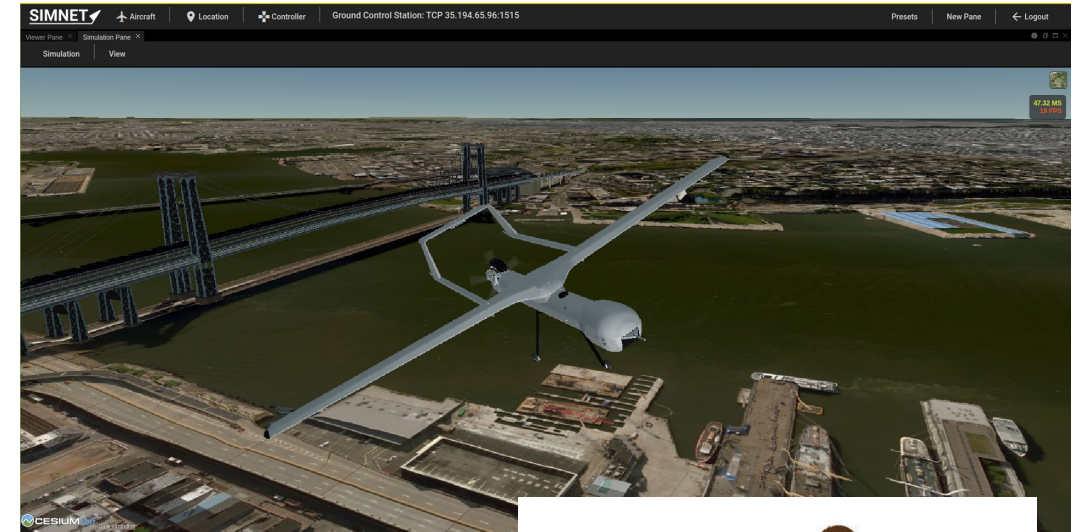
- ▶ The purpose of this presentation is to introduce benefits, best practices and examples of simulation software in online course environments.
- ▶ The research team does not have any affiliation with the introduced software platforms.

# Overview

- ▶ 1. Background and conditions
- ▶ 2. Purpose and integration
- ▶ 3. Examples and best practices
- ▶ 4. Lessons learned and outlook

# Benefits of Simulation Platforms

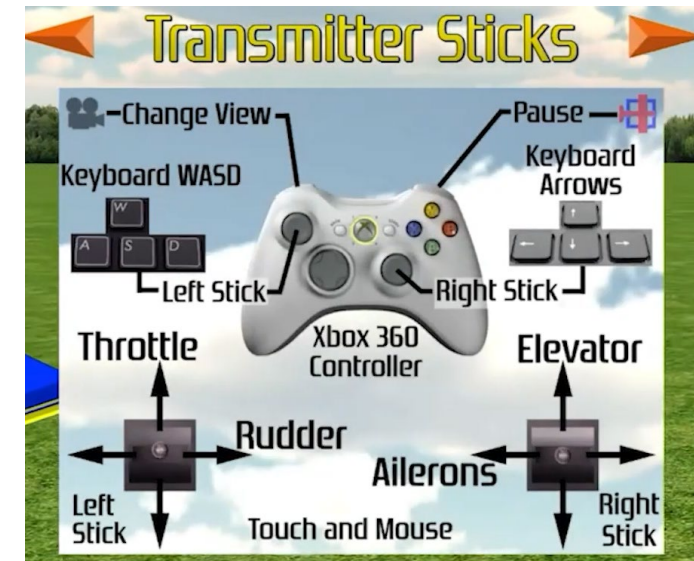
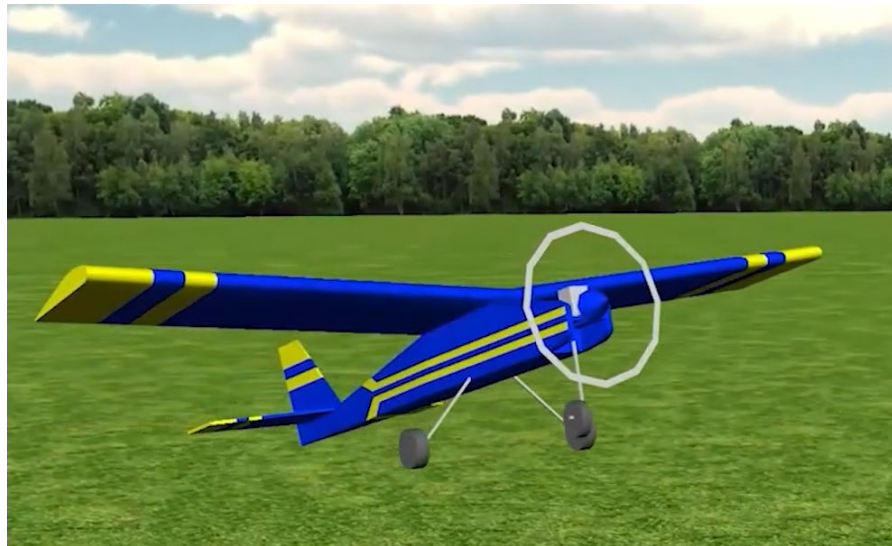
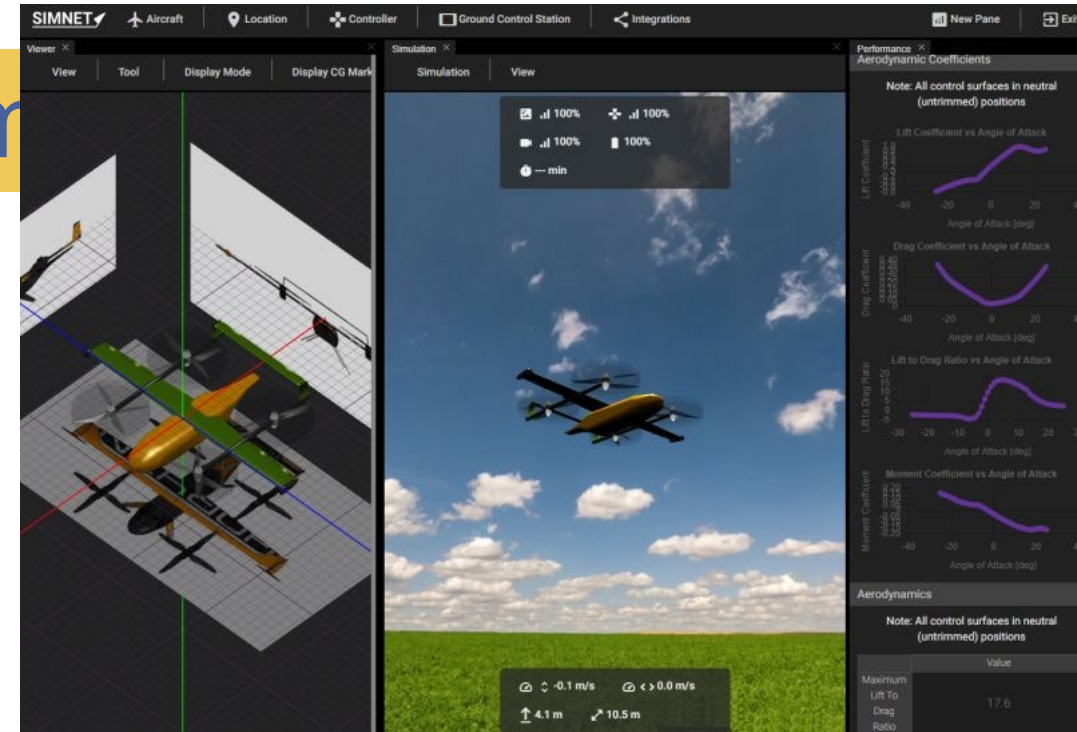
- ▶ Inquiry based learning
- ▶ Constructivist learning
- ▶ Creativity





# Benefits of Simulation Platform

- ▶ Visualization
- ▶ Problem solving; and
- ▶ Challengebased projects



# Conditions and Vetting Parameters

- ▶ Availability (region, OS)
- ▶ Applicability
- ▶ Download package or browser based
- ▶ Onboarding and tutorials
- ▶ Affordability– Open Source or free trials available?
- ▶ Affordability– create accounts and purchasing models (e.g. 3 month license)
- ▶ Integrate in Master Textbook List as course material

# Conditions and Vetting

- ▶ Coordinate with Academic Technology Dept. for evaluation
- ▶ Structured process for "request for a tool review" with AT
- ▶ Utilization as course material external to Canvas




# Conditions and Vetting


## ▶ Activity/assignment creation

- Home
- Syllabus
- Announcements
- Modules
- Discussions
- Grades
- People
- EagleVision Zoom
- New Analytics
- Course Evaluation Administrator
- Collaborations 
- Quizzes 
- Outcomes 
- Pages 
- Files 
- Assignments** 
- Summary 
- Item Banks

### 1.4 SIMNET Introductory Lab

 Assignment

In this activity, you will become familiar with the SIMNET workspace and complete a short lab assignment followed by a compare/contrast report on the three types of sUAS covered in the lab.



Before participating in this lab, you should have completed the required reading in the Reading and Resources section.

#### Account

You should have a SIMNET account set up as part of the required materials for this course. **If you have not already set up your account and reviewed the introductory video**, please do so per the instructions below:

1. Access SIMNET, and complete your user profile.  
Note: If you have not already created your SIMNET account, the directions are provided for you here:
  - i. SIMNET requires the use of the Chrome browser. Install [Chrome](#) from Google, if needed.
  - ii. Create a new SIMNET account through [SIMNET Sign Up](#).
2. Review the [SIMNET Tutorial: Quickstart Guide \(6:57/YouTube\)](#) to become familiar with SIMNET and its features.

#### Lab

Now that you are familiar with SIMNET, enter the workspace. Perform the following tasks to develop a better understanding of the capabilities and differences between Multicopter, Fixed-Wing, and VTOL type sUAS:

Load the following aircraft into the SIMNET workspace, and use the values provided by the Performance Pane to fill the table below.

# The Environment Setting the Scene

- ▶ Using screenshots and recordings as proof and evidence
- ▶ Using individual and group activities
- ▶ Sharing of results and peer review
- ▶ Creativity fosters the achievement of CLO's

# The Environment Setting the Scene

► Using screenshots and recordings as proof and evidence

## Lab

Now that you are familiar with SIMNET, enter the workspace. Perform the following tasks to develop a better understanding of the capabilities and differences between Multicopter, Fixed-Wing, and VTOL type sUAS:

Load the following aircraft into the SIMNET workspace, and use the values provided by the Performance Pane to fill the table below.

Aircraft	UAS Type	Endurance [minutes]	Maximum Range [km]	Maximum Speed [km/h]
H140 Hexacopter	Multicopter			
Q400 Fixed Wing UAV	Fixed-Wing			
Q400 VTOL UAV	VTOL			

Prepare a one to two-page report that includes the results from the table above and compares the strengths and weaknesses of each UAS type in terms of performance. Tie this information in with the information presented in your readings about each UAS type in terms of its unique capabilities and/or shortcomings.

Your lab report may include screenshots if desired and does not need to be in APA format **with the exception of references provided at the end**. Be sure to review the SIMNET Lab Rubric to understand expectations for your

The screenshot displays the SIMNET software interface. On the left, a 'Designer' pane shows aircraft specifications: Weight (24), Battery Cell Count (24), Payload Type (Payload Mass), Payload Mass (1.32), Inj. (No Warnings), and Design Warnings (No Warnings). Below this is the 'Recommended Design Specifications' section with values for Airframe Size (1.067), Aircraft Size (1.753), All Up Weight (11.8), Motor (40 KV 7 Amps 717 Watt), ESC (9 Amps 24S), Propeller (27.0 x 8.1), and Battery (24S1P 11367 mAh 20 C). The 'Predicted Performance' section shows Hover Time (Full Capacity) at 62.9 and Hover Time (80% Capacity) at 50.3. The central 'Viewer' pane shows a 3D model of a hexacopter. On the right, the 'Performance' pane displays graphs for 'Flight Time vs Speed' and 'Range vs Speed', along with a table of predicted performance values: Max Range (28.8), Cruise Speed for Max Range (16.1), and Angle of Attack for Max Range (-11.2).

The screenshot shows a notification screen with a blue ribbon icon containing a checkmark. The text reads 'Passed Training Exercise' at the top, 'Training Exercise Passed' in the center, and 'Exercise: SIMNET Training MC-A-001' at the bottom. The background is a blurred aerial view of a city.

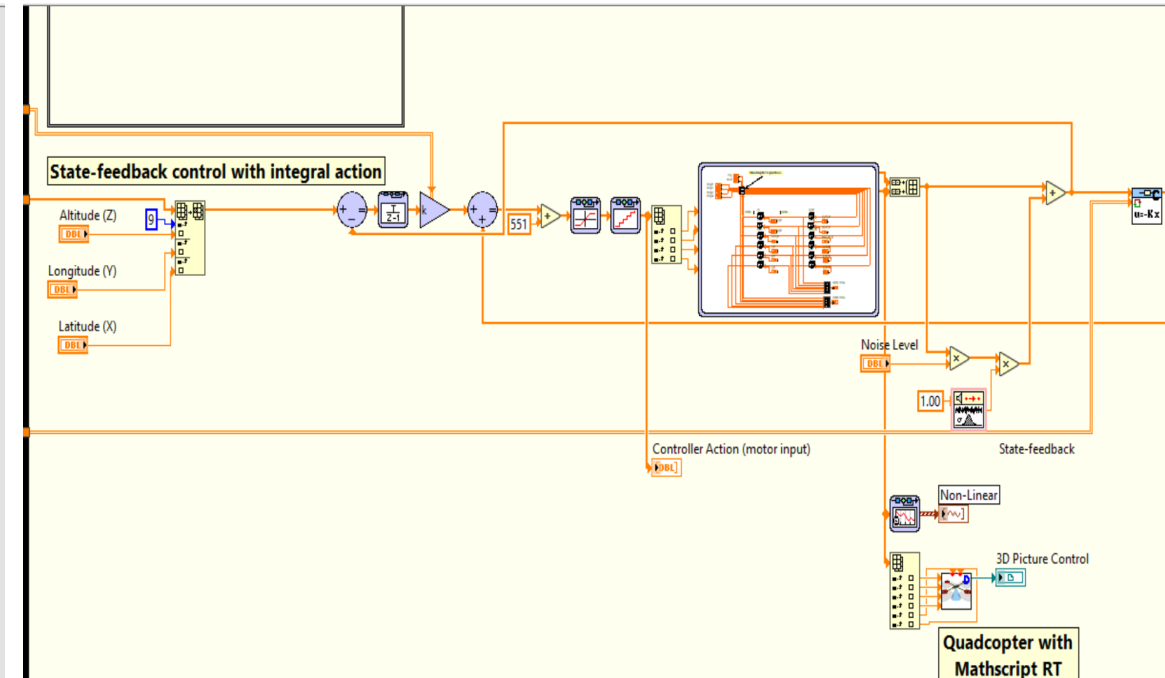
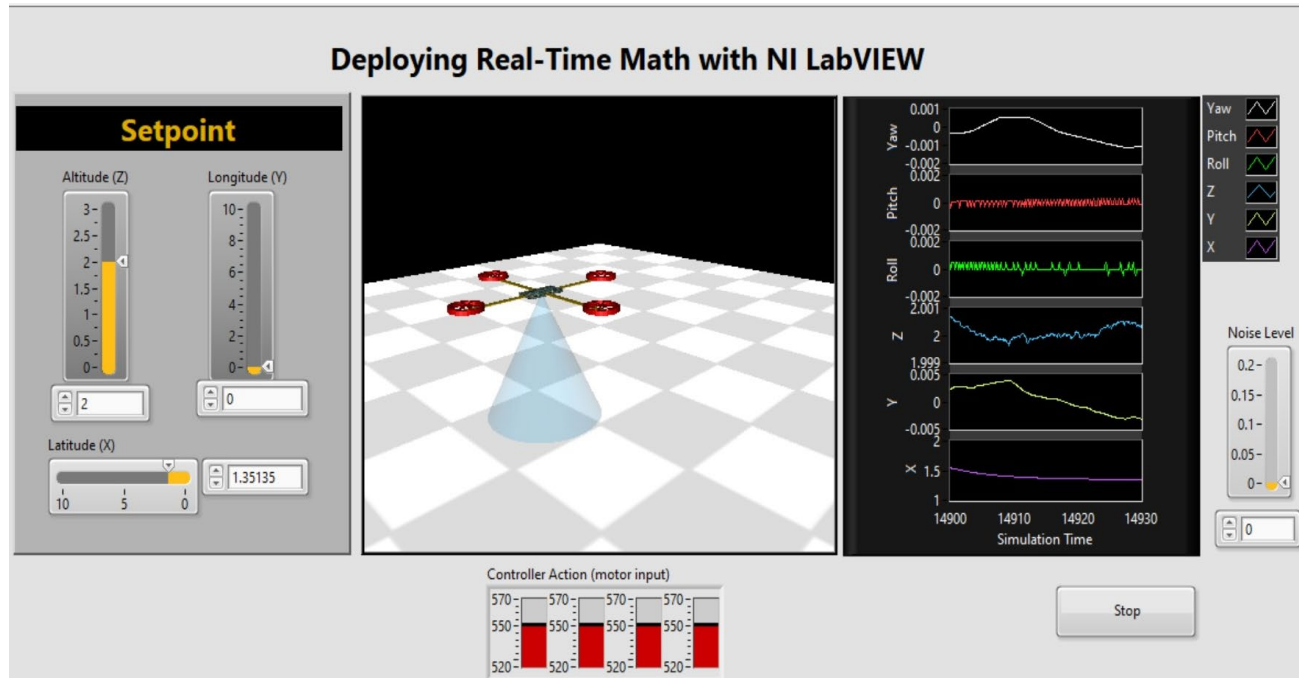
# Introduction of Simulation Software

- ▶ LabVIEW
- ▶ MATLAB and SIMULINK
- ▶ SIMUAID/LOGICAID/VIVADO
- ▶ Multisim
  
- ▶ SIMNET AERO
- ▶ WeBots
- ▶ AutoCAD Fusion

# Simulation Software for Engineering

# LabVIEW

## ► Quadcopter Dynamics and Control



# MATLAB/SIMULINK

The image displays the MATLAB/Simulink environment. On the left, a 3D visualization of a quadcopter drone is shown in a virtual environment. The drone is positioned over a green field with a grey path. The interface includes a menu bar (File, View, Viewpoints, Navigation, Rendering, Simulation, Recording) and a toolbar with various simulation controls. A status bar at the bottom of the 3D view shows the current time (T=21.20) and position (Pos:[57.90 2.17 95.99] Dir:[-2.44 -1.6]).

On the right, the Simulink workspace is open, showing a block diagram titled "Quadcopter Flight Simulation Model - Mambo". The diagram consists of several interconnected blocks:

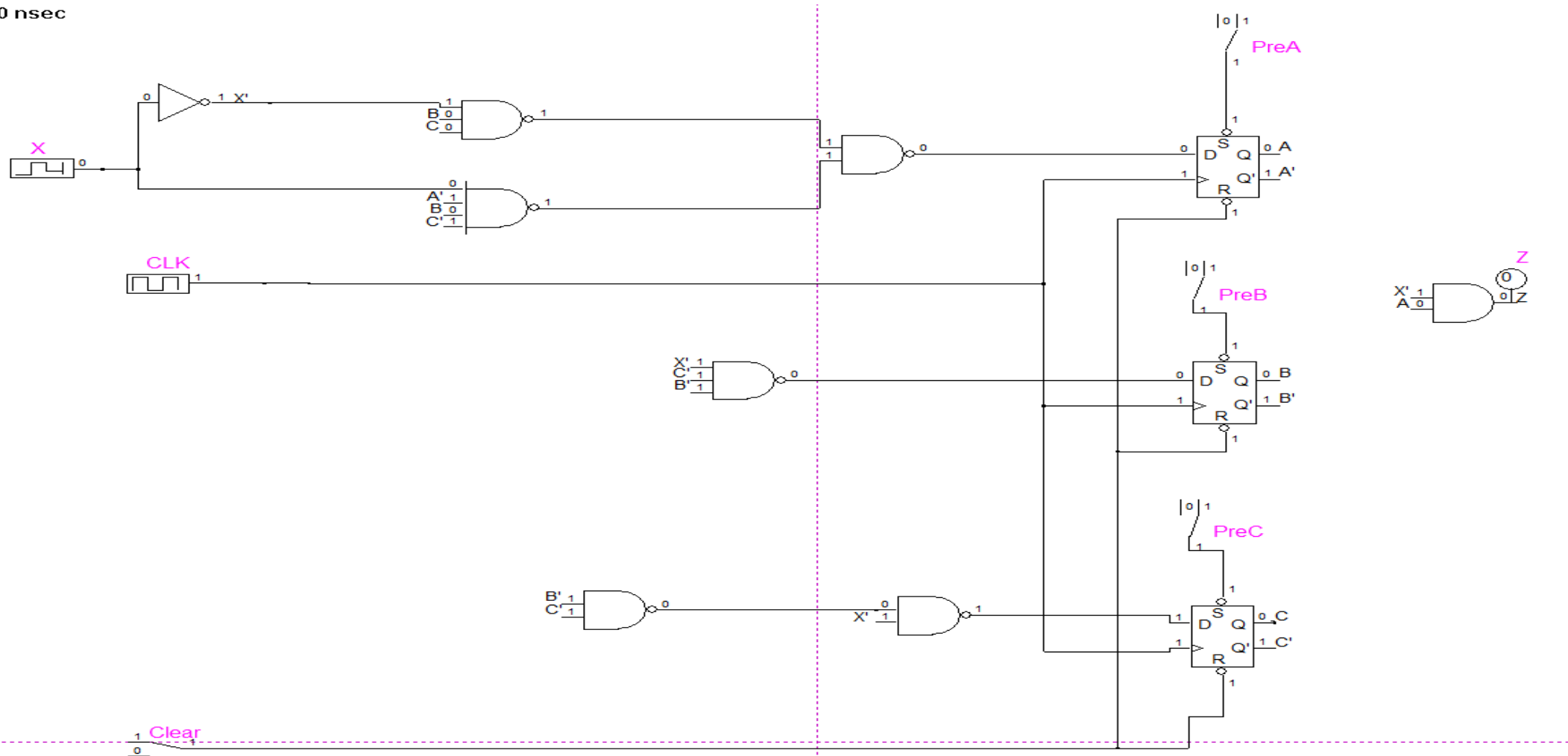
- Signal Editor AC Cmd**: Provides a Command input to the flightControlSystem.
- Sensors (Dynamics)**: Receives States and Environment inputs and outputs Sensors and Image Data.
- flightControlSystem**: Receives Command and Sensors inputs and outputs Actuators and a Flag.
- Nonlinear Airframe**: Receives Actuators and Environment inputs and outputs States.
- Commands**: Receives Actuators and States inputs and outputs Visualize.
- Flags**: A red block that outputs Stop Simulation.

The Simulink interface includes a menu bar (SIMULATION, DEBUG, MODELING, FORMAT, APPS) and a toolbar with simulation controls (Step Back, Pause, Step Forward, Stop, REVIEW RESULTS). The workspace title is "asbQuadcopter". The status bar at the bottom indicates the simulation is "Running" at 100% completion, with a time of T=21.420 and a progress bar at 71%.



# Digital Circuit Simulation Software SIMUaid

0 nsec



# Digital Circuit Simulation Software Vivado

The screenshot displays the Vivado IDE interface during a behavioral simulation. The main window shows a timing diagram for a 16-bit shift register. A yellow vertical line marks the current simulation time at 782.000 ns. The timing diagram shows the clock (CLK) and clear signals, along with the 16-bit output (Sout[15:0]). The output values are shown in hexadecimal: c000, e000, 0000, 8000, c000, 6000, 3000, 9800, 4c00, 2600. The simulation time is 21 us.

**Flow Navigator**

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION**
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design

**Simulation Window: Untitled 1**

Scope: Sources

Name	Value
CLK	1
Clear	1
Sin	1
Sout[15:0]	e000
[15]	1
[14]	1
[13]	1
[12]	0
[11]	0
[10]	0
[9]	0
[8]	0
[7]	0
[6]	0

Protocol Instances

Simulation Time: 21 us

# Circuit Simulation Software Multisim

The screenshot displays the Multisim software interface for a circuit simulation. The main workspace shows a common-emitter amplifier circuit. The circuit components and their values are:

- VCC: 9V
- RB\_1: 100kΩ
- RB\_2: 47kΩ
- RC: 680Ω
- RE: 330Ω
- CE: 470µF
- CE1: 220µF
- RL: 1kΩ
- Cin: 10µF
- Q1: 2N2222A

An oscilloscope (XSC1) is connected to the collector and emitter nodes. The oscilloscope display shows a sine wave signal. The oscilloscope control panel includes the following data:

Time	Channel_A	Channel_B
T1	87.812 ms	9.951 mV
T2	87.812 ms	-716.527 mV
T2-T1	0.000 s	0.000 V

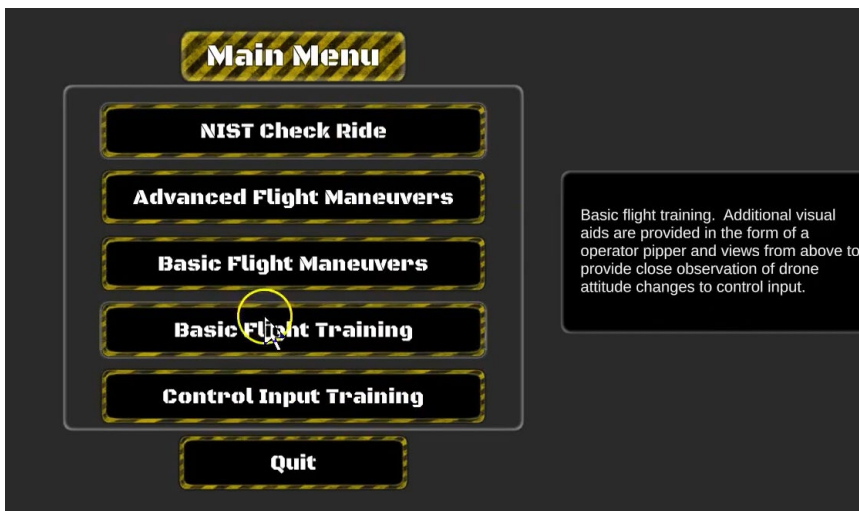
The software interface also shows a menu bar (File, Edit, View, Place, Simulate, Transfer, Tools, Reports, Options, Window, Help), a toolbar, and a component library on the left side.

# UAS Simulation Preparation

- ▶ Allows for effective operation and application of sUAS.
- ▶ Introduced via our Dept. Of Flight in the sUAS Ops Minor
- ▶ Supports learning of sUAS systems and operating requirements as well as navigational competencies.
- ▶ Offers practice on scenario based modeling to improve students' airmanship skills and understanding of course material.

\*See AIR RT Session "Integrating VR into the Asynchronous Learning Environment" Dr. Sanders, Dr. Marchant, Dr. Thirtyacre, Mr. Delcastillo

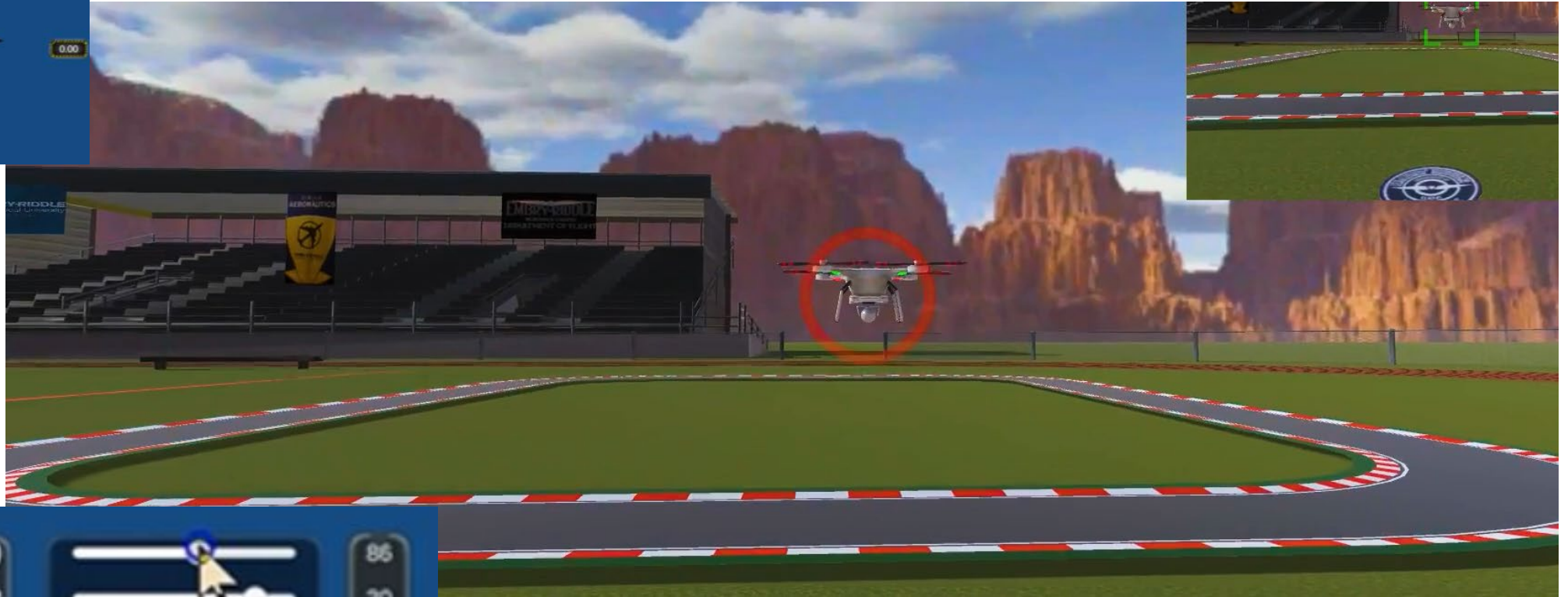
# Simulation Preparation



ERUPTSim was ERAU WW developed. The flight simulator (Unity-engine) offers training remote pilots of drones/sUAS.



# Simulation Preparation



Max Yaw Velocity (deg/sec) 86  
Max Vehicle Pitch/Roll Angle 30  
Dead Zone Left Stick 0.00  
Dead Zone Right Stick 0.00  
Smoothing 0.10

A control panel with five sliders. The first slider is labeled 'Max Yaw Velocity (deg/sec)' and has a value of 86. The second slider is labeled 'Max Vehicle Pitch/Roll Angle' and has a value of 30. The third slider is labeled 'Dead Zone Left Stick' and has a value of 0.00. The fourth slider is labeled 'Dead Zone Right Stick' and has a value of 0.00. The fifth slider is labeled 'Smoothing' and has a value of 0.10. A mouse cursor is pointing at the first slider.

ERUPTSim: Works on computer, laptop and in VR environment (Oculus Quest) and is compatible with Windows and Mac.

# SIMNET AERO





# SIMNET AERO

- ▶ Design, analysis, and simulation of multicopter, fixed wing, and VTOL drones.
- ▶ Configurator– create, design and engineer an sUAS.
- ▶ Simulator– sUAS flight simulation with QGroundControl.
- ▶ Mission Planning.



# SIMNET AERO

The screenshot displays the SIMNET AERO software interface. The main window is divided into a 3D simulation pane on the left and a configuration panel on the right. The 3D pane shows a black propeller model on a grid, with a coordinate system (X, Y, Z) and axes (Red, Blue, Green) visible. The configuration panel on the right is titled 'Propeller' and contains several sections: 'Info', 'Mass', 'Position', 'Appearance', and 'Design'. Each section contains a table of parameters and their values.

Info	
Name	Propeller
Description	

Mass	
Mass [kg]	0.093
Mass Margin [%]	0

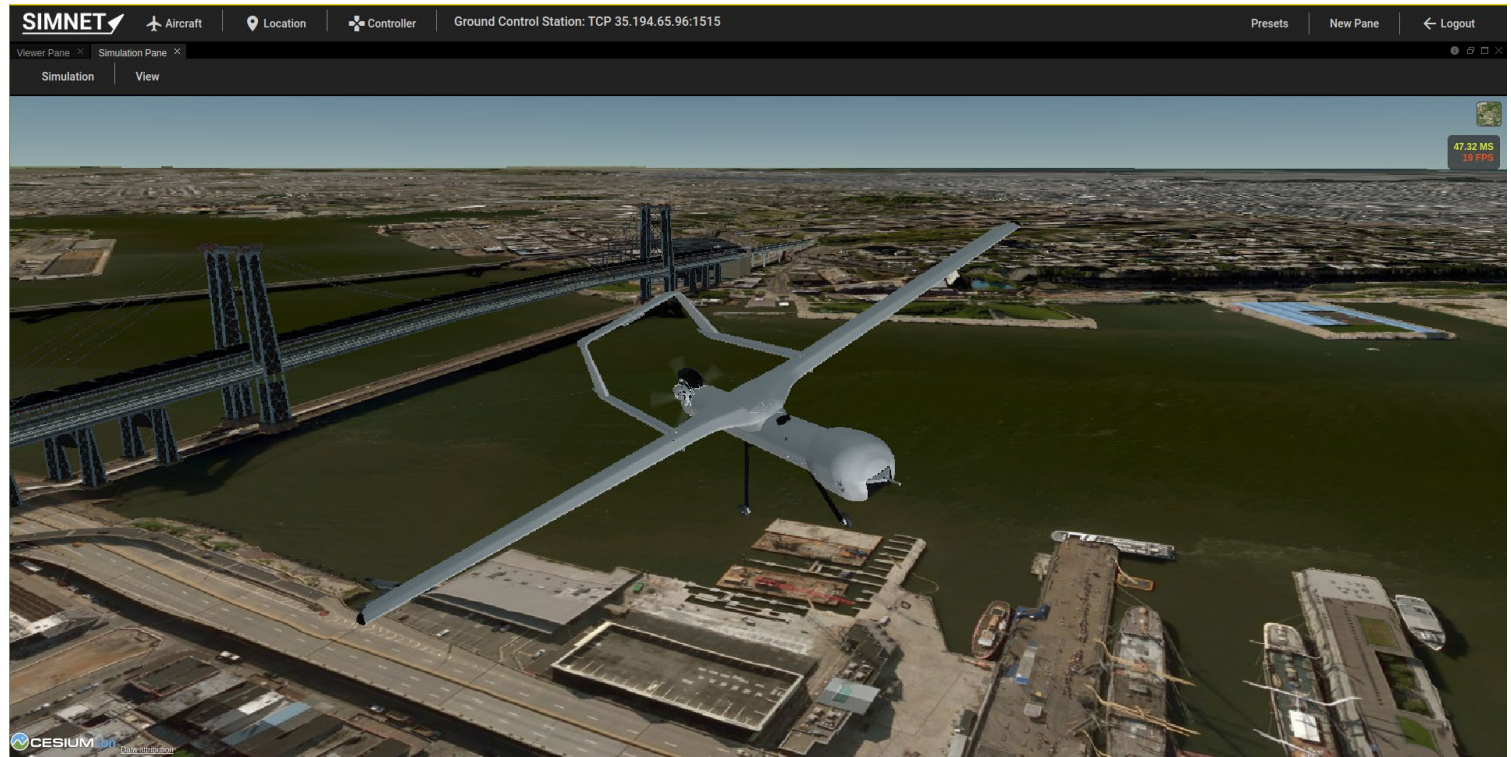
Position		
	Position [m]	Rotation [deg]
X/Roll/Front/Red Axis	0	0
Y/Pitch/Right/Blue Axis	-0.764	0
Z/Yaw/Down/Green Axis	0.295	180

Appearance	
Red Color [%]	40
Green Color [%]	40
Blue Color [%]	40

Design	
Select from Database	APC_SlowFlyer
Width [cm]	
Diameter [in]	19
Pitch [in]	15

# SIMNET AERO

- ▶ Works as course material in BSUSA
- ▶ 3-month software license
- ▶ purchased through the bookstore



# AutoDesk Fusion 360



*Freelicensekeys.info*  
AUTODESK®  
FUSION 360™



# AutoDesk Fusion 360

- ▶ CAD – Computer Aided Design
- ▶ Portfolio of AutoDesk
- ▶ Tool for object generation, engineering, and industrial design
- ▶ Students and educators can get free one year educational access to Autodesk products and services, renewable as long as you remain eligible.



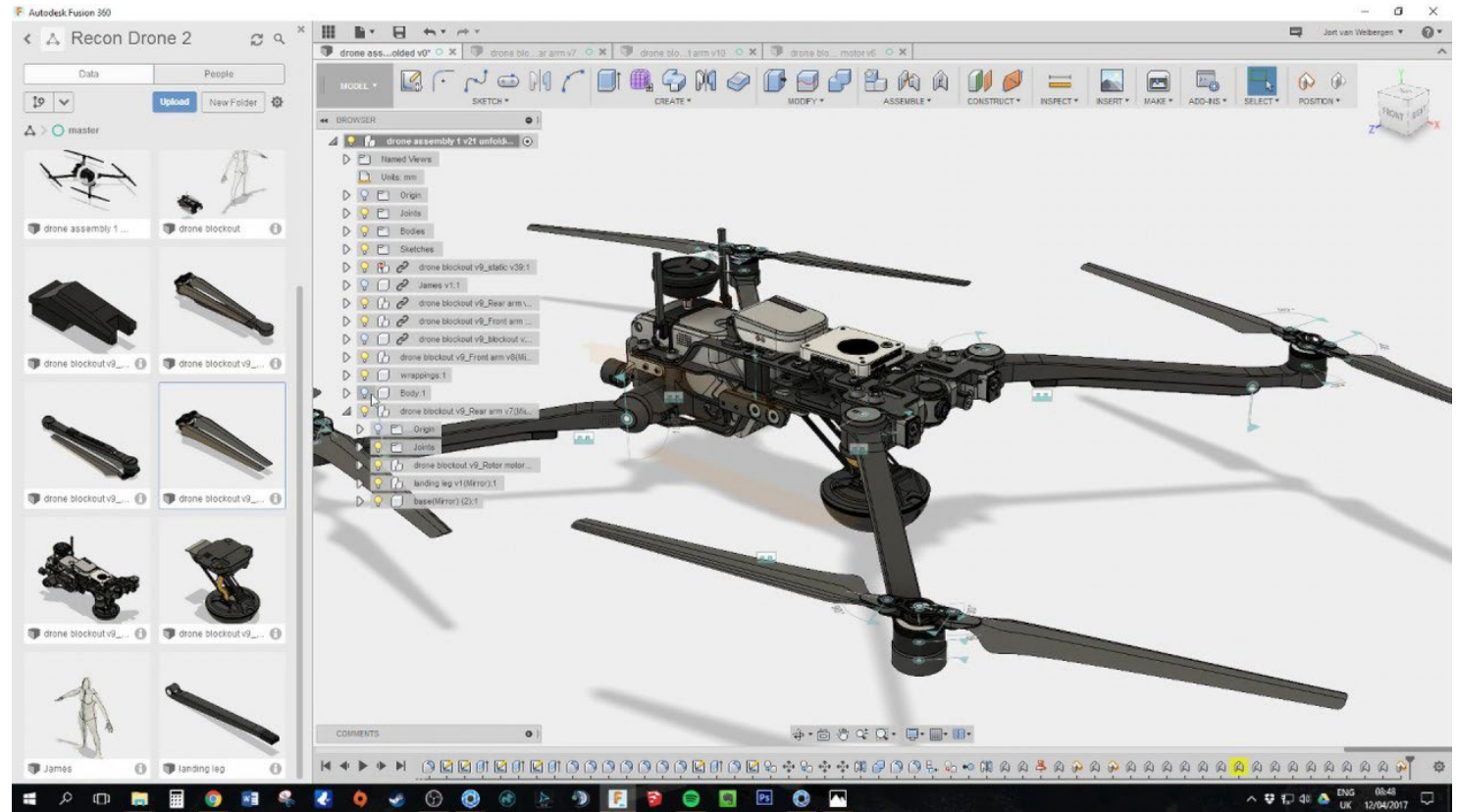
# AutoCAD Fusion 360

- ▶ Forms
- ▶ Sizes
- ▶ Joints
- ▶ Actuators
- ▶ Locomotion
- ▶ Sensors



# AutoCAD Fusion 360

- ▶ Uncrewed and robotic systems
- ▶ All Domains
- ▶ Shapes
- ▶ Elements
- ▶ Sizes





# Cyberbotics WeBots Robots in Virtual Worlds



# Cyberbotics WeBots Robots in Virtual Worlds

The screenshot displays a virtual world simulation. The central view shows a WeBot robot, a small four-wheeled vehicle with a yellow and black body and a brown cardboard box on top, positioned on a brown, uneven terrain. The interface includes a top toolbar with playback controls and a timer showing 0:01:40:620 at 0.85x speed. On the left, a scene tree lists various objects like WorldInfo, Viewpoint, TexturedBackground, and several instances of Sassafras trees and Oak trees. At the bottom left, a console window shows the following text:

```
Console - All
Recording at 25 FPS, 1054687 bit/s.
Video encoding stage 1... (please wait)
13th target reached
Video encoding stage 2... (please wait)
INFO: Video creation finished.
1st target reached
2nd target reached
```

On the right, a code editor window titled `..wing/moose_path_following.c` shows the following C code:

```
1 /*
2  * Copyright 1996-2021 Cyberb
3  *
4  * Licensed under the Apache
5  * you may not use this file
6  * You may obtain a copy of t
7  *
8  * http://www.apache.org/
9  *
10 * Unless required by applica
11 * distributed under the Lice
12 * WITHOUT WARRANTIES OR CONL
13 * See the License for the sp
14 * limitations under the Lice
15 */
16
17 #include <math.h>
18 #include <stdio.h>
19 #include <webots/compass.h>
20 #include <webots/gps.h>
21 #include <webots/keyboard.h>
22 #include <webots/motor.h>
23 #include <webots/robot.h>
24
25 #define TIME_STEP 16
26 #define TARGET_POINTS_SIZE 13
27 #define DISTANCE_TOLERANCE 1.
28 #define MAX_SPEED 7.0
29 #define TURN_COEFFICIENT 4.0
30
31 enum XYZComponents { X = 0,
32                    Y = 1, Z = 2 };
33
34 typedef struct _Vector {
35     double u;
36     double v;
37 } Vector;
38
39 static WbDeviceTag motors[8];
40 static WbDeviceTag gps;
41 static WbDeviceTag compass;
42
43 static Vector targets[TARGET_POINTS_SIZE];
44 {-4.209318, 9.147717}, {0
45 {-4.394915, -24.550777}, {-
46 {0.175989, -1.784311}, {0
47
48 };
49 static int current_target_index = 0;
50 static bool autopilot = true;
51 static bool old_autopilot = true;
52 static int old_target_index = 0;
```

# Cyberbotics WeBots Robots in Virtual Worlds

- ▶ WeBots is an open source threedimensional mobile robot simulator.
- ▶ It was originally developed as a research tool to investigate various control algorithms in mobile robotics.
- ▶ minimal knowledge is needed in mobile robotics, in C, C++, Java, Python or MATLAB programming, and in VRML97 (Virtual Reality Modeling Language).
- ▶ Free for download.
- ▶ Excellent User Guide for Robotics Introduction.

# Cyberbotics WeBots Robots in Virtual Worlds



# Future Outlook

- ▶ Drone technologies will increasingly be used as a tool to enhance teaching and learning effectiveness.
- ▶ New software will increase student engagement in project-based learning in STEM education to address programming aspects and application of UAS for the collection of remote sensing data.
- ▶ Additional considerations for faculty training and professional development will be needed to address training on new systems as well as ongoing changes in policy and regulations.

# Lessons Learned

- ▶ Very good feedback from students
- ▶ Engaging and creative learning environment
- ▶ Students can explore, trial, and improve
- ▶ Compare results in the classroom or discussion board
  
- ▶ **Constructivist learning is an excellent plagiarism prevention.**



Thank you