125 YEARS AT THE HEART OF ENGINEERING EDUCATION
ASEE
2018 ASEE ANNUAL CONFERENCE & EXPOSITION
JUNE 24–27, 2018
SALT PALACE CONVENTION CENTER
SALT LAKE CITY, UT

Paper ID #22143

Robotics in Electronics Engineering Technology

Dr. Asad Yousuf, Savannah State University

Asad Yousuf is the Coordinator and Professor of Electronics Engineering Technology at Savannah State University

Dr. Mohamad A. Mustafa, Savannah State University

Mohamad Mustafa is a Professor of Civil Engineering Technology and the Chair of the Engineering Technology Department at Savannah State University (SSU). He has six years of industrial experience prior to teaching at SSU. He received his BS, MS, and PhD in Civil Engineering from Wayne State University, Detroit, Michigan.

Mr. Alberto G. De La Cruz
Alfredo Villanueva
Dr. Mir M. Hayder, Savannah State University

Dr. Hayder is an Assistant Professor in the Department of Engineering Technology at savannah State University, GA. He received PhD in Mechanical Engineering from McGill University, Canada. His research interest lies in the areas of fluid-structure interaction, flow-induced vibrations, syngas and blended fuel combustion, and flow and structural simulations, robotics and STEM education.

Robotics in Electronics Engineering Technology

Abstract

Over the last several years we have developed curriculum to support Electronic Engineering Technology (EET) students using Robotics in projects. We have had an enthusiastic response from students interested in both Robotic projects and interest in learning more about Robotics. Robotics is an interdisciplinary field that incorporates the integration of many systems in software, electronics, control systems, actuators and sensors. The Robotics of today imparts the most important attributes such as the nature of motion, the motions available to rigid bodies and the use of kinematic constraints to organize motion. Because the growing field of Robotics covers many areas of EET education we decided we would develop curriculum for an introductory course in Robotics. This paper explores the curriculum design and the Lie Algebra and Lie Group that are keeping track of the variables involved in arm robotic movement. These mathematical aspects are presented in a simplistic approach for the EET students. A survey of Introductory Robotic courses on the internet showed us that 2D and 3D representations of Robot pose, rotations using quaternions, Kinematic Chains, Forward and Inverse Kinematics, Dynamics, actuators and sensors were main topics of the courses. We intend to present our design of the course to cover those topics geared for EET students with special emphasis on control systems, actuators and sensors. The course splits between weekly assignments on theory and weekly assignments using student built Robot Arm, mobile Robot, and simulated Robots using the Robotic Operating System (ROS). Projects will be progressive starting with micro-controller Robot based Systems to more complex Robotic Systems using ROS.

Introduction

The Robotics Industry is expected to grow at "compound annual growth rate (CAGR) in the teens or greater" through 2022 [1].

Over the past several years we have developed Robotic Labs and projects that we have incorporated into other Electronic Engineering Technology (EET) courses to motivate students in those courses. We noted high student motivation on those Labs and projects, as well as positive feedback from students.
Robotics is an interdisciplinary field requiring knowledge of software, electronics, control systems, actuators and sensors.

After reviewing course materials from courses on-line and number of introductory books, we determined that an introductory course for Electronic Engineering Technology (EET) should cover the following topics:

1) Robots and Degrees of Freedom
2) Representation of Rigid-Bodies in 2D and 3D Space
3) Kinematic Chains and Forward Kinematics
4) Velocity Kinematics and Statics
5) Inverse Kinematics
6) Dynamics

7) Control Systems, Actuators and Sensors

Actuators, Sensors and Control Systems are important elements of the EET program. We made a point to emphasize these areas of Robotics in the Laboratory section of the course.

Course Layout

We felt the textbook that best fit our needs was "Modern Robotics Mechanics, Planning and Control" by Kevin M. Lynch and Frank C. Park. One of the many advantages of the textbook is short video lecture segments provided on-line for students to view. The videos covered important areas of the first eight chapters [2]. We also suggested students use "A Mathematical Introduction to. Robotic Manipulation" as a supplementary text to the "Modern Robotics" textbook. Appendix A, Lie Groups and Robotic Kinematics of the textbook gives a rigorous introduction to Lie and algebra [3]-[5].

The "Modern Robotics" text book covers Lie groups and algebra as well as associated exponential mathematics (see Table 1):

Lie Group	Lie Algebra
SO(2)	so(2)
SE(2)	se(2)
SO(3)	so(3)
SE(3)	se(3)

Quaternions and Euler angles mathematics is covered in Lab 5 of the course.

We flipped the Lectures for chapters 2 through 8 with the students viewing the videos and doing the homework assignments. Class time for chapters 2 through 8 were used to work homework problems with students and answer student questions. Classes for Control Systems, focus on Velocity, Torque, Force and Impedance control of the Robot.

Labs

The Electronics Lab has 10 workstations running Windows. In addition to the workstations there is a laptop running Linux and ROS. Each workstation has the Ardunio Development Environment, along with Matlab, Simulink and V-REP installed. The micro-controllers used by both the mobile and arm robots are the Ardunio and MBED.

The labs are split into four overlapping areas:
a) Student Built Robot Arm
b) Student Built Mobile Robot
c) Simulations with Coppelia Robotics V-REP PRO EDU
d) Robotic Operating System (ROS)

Labs are split equally between Non-Mobile Robots and Mobile Robots using the above systems. Table 1 shows the list of Labs for the course by week.

#	Lab Type	Description
1	Non-Mobile	Students build a 3R robot with RC Servo Motors and control with µC
2	Non-Mobile	Controlling 3R robot with Matlab and Forward Kinematics Routine
3	Both	V-REP Simulator Introduction
4a	Non-Mobile	Inverse Kinematics with V-REP and 3R Robot
4b	Mobile	ROS Control of Drone
5	Non-Mobile	Rotations with Euler Angles and Quaternions using ROS/Matlab
6	Non-Mobile	Exploring D-H Parameters using V-REP and Matlab
7	Mobile	Serial Communication between Linux Laptop and ESSUR2 Robot Base
8	Mobile	Controlling VEX DC motor with Encoders on ESSUR2 Robot
9	Mobile	PID and Limits to Acceleration/Deceleration
10	Mobile	Using IMU with µC on ESSUR2 Robot
11	Mobile	Data Fusion
12	Mobile	Controlling ESSUR2 Robot from ROS
13	Mobile	Open CV and Drone

Table 1 List of Labs

Non-Mobile Robots

In the first Lab students build a 3R Robot using simple materials, 3 RC Servo Motors (0-180°) and an Arduino micro-controller. The serial link and Arduino Development Environment is used to control the Robot. A hook at the tip of the Robot is used to move light weight paper forms [6].

Lab 2 uses MATLAB to control the 3R Robot over the Serial Port. The Forward Kinematics from the Modern Robotics book is used to calculate the position of the Robot tip given the joint angles. The actual position of the Robot is measured and compared to the calculated position to determine the accuracy of the Robot [7].

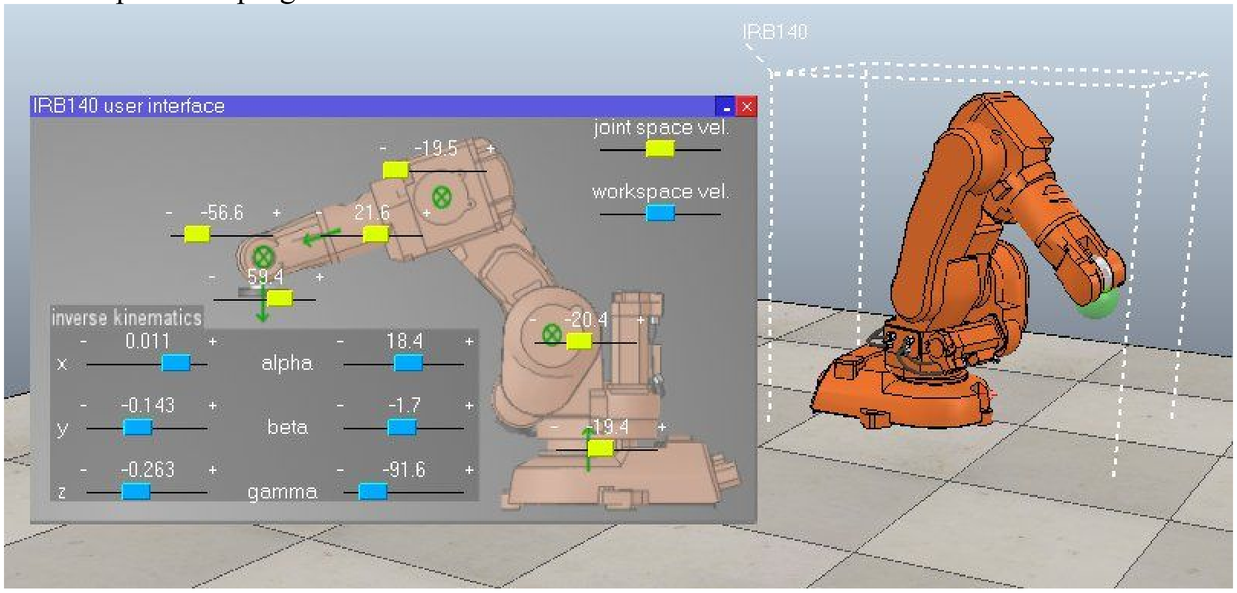
Lab 3 introduces the main features of the V-REP Simulator:

a) Scene Objects
b) Calculation Modules
c) Control Mechanisms

The model browser comes with a wide variety of Robot models both mobile and non-mobile.

The students are provided V-REP scripts that demonstrate the major features of V-REP Simulator [8].

The LUA programming language is also introduced to the students. The students write and run some simple LUA programs such as “Hello World”.



Inverse Kinematics is at the center of Lab 4a using first V-REP and then the 3R Robot. The ABB IRB 140 Robot Model in V-REP is inserted into a scene and the simulator is run. The ABB IRB 140 simulation has a widget that allows the students to input joint angles or end effector position to control the Robot Arm. The 6R ABB IRB with the V-REP simulation running is shown in Figure 1.

Figure 1 IRB140 CAD Robot model is courtesy of ABB Robotics. ABB Robotics has however no link to V-REP

Students use the model to explore the limits of the workspace and singularities. Using the V-REP Simulator, students find the joint angles needed to achieve the desired positions for the 6R Robot. Students then use the inverse kinematics routine position in the Modern Robotics toolbox to find the required joint angles for the 3R Robot to move to required positions given in the Lab. The position of the 3R Robots end effector is physically measured to insure it is in the correct position, using the joint angles predicted by the inverse kinematics routine.

In Lab 4b, students will be using ROS Kinect Kame to control basic movements of a drone. Students will write python scripts to have the drone move autonomously forward and backward. This teaches the ROS fundamentals of publishing and subscribing to nodes in ROS. Fundamentals of the available ROS topics are provided in the bebop_autonomy repository, which will be used to make basic connections to a Parrot Bebop 2 drone.

http://bebop-autonomy.readthedocs.io/en/latest/installation.html

In Lab 5 students use ROS and python to convert between Euler angles and quaternions. ROS is used to visualize an aircraft model to understand how Euler angles work as well of their limitations with the Gimbel Lock issues. Students write a routine in Matlab to perform quaternion multiplication and find the inverse of a quaternion [9]. Students use the routines they wrote to calculate orientation and position of links in a simple Simulated Robot.

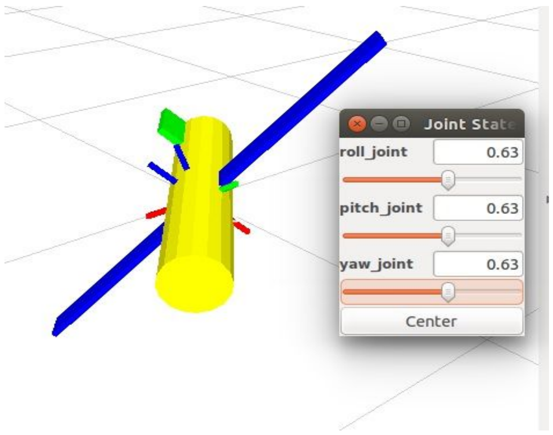


Figure 2 Aircraft shown in ROS RVIS Simulator

Figure 2 shows the aircraft model and the slider window for Euler angles (Roll, Pitch, Yaw). The RVIS simulator displays the quaternion representation for Roll, Pitch and Yaw, so students can check there conversion of Euler Angles to quaternions and quaternions to Euler Angles are correct.

D-H parameters are explored in Lab 6 using the V-REP Simulator. In the first part of the Lab students are given the specification of a Robot Arm and asked to fill in the D-H table entries. In the second part of the Lab students use the D-H tool in V-REP to find the D-H parameters for the given Robot Arms. Students must explain why they matched or why they are different. Finally students use Matlab to calculate an end effector for a given Robot Arm and compared to calculated position using se(3) [10], [11].

Mobile Robots

The Mobile Robot Labs 7-12 use the Environmental Savannah State University Robot 2 (ESSUR2) Robot gives students the opportunity to learn about some of the components typically found in a Robot Base [12]. An Arduino Uno board and MBED board combination provide the link between ESSUR2 Robot Base sensors and actuators and a Linux Laptop that is mounted on the ESSUR2 Robots top deck. Figure 3 is a Block Diagram of the ESSUR2 Robot Base. The Ardunio is connected through a I2C interface to a 9 Degree of Freedom (DOF) Sensor Stick from Sparkfun Inc. The Arduino and Sensor Stick are used as a Inertial Measurement Unit (IMU) for the Robot Base. The Sensor Stick uses a LSM9DS1 which provides 3 axis accelerometers and 3 axis Gyros and 3 axis Magnetometers for a total of 9 DOF [13].

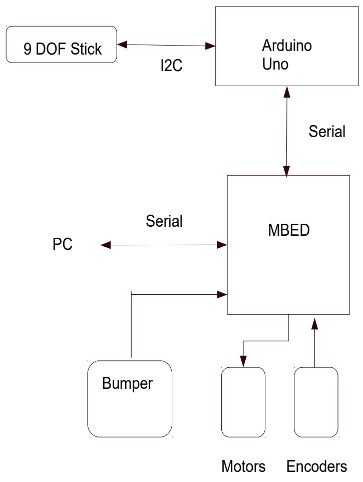


Fig. 3 Block Diagram ESSUR2 Robot Base

The Arduino Uno is connected to the MBED through a Serial Interface. The MBED detects the Bumper switches connected to the Robot Base and the Left/Right wheel encoders are connected to the front wheel axles. The Left/Right VEX DC Motors are connected to VEX Motor Controllers. The VEX Motor Controllers are connected to ports on the MBED. The MBED uses two PWM signals to control the Left/Right Motors Controllers and DC Motors. Finally the Robot Base is controlled by a USB Serial Interface to the Linux Laptop mounted on top of the ESSUR2 Robot Base.

To assist the students writing and testing code within a two hour Lab, students are given code stubs to work from which define the I/O and processing required for the routine.

For Lab 7 students establish and test the serial interface between the MBED and Linux Laptop to control the ESSUR2 Robot. Students will code and test in the MBED for the Status Function of the Interface Protocol for the Serial to/from the PC/Laptop.

Students in Lab 8 code and test the routines in the MBED that control the Vex Motors speed and read the quadrature VEX Encoder [14], [15]. The test consist of moving the Robot a set distance in a straight line and rotating the Robot about the center point of the front wheel axis. Note the Encoder is not used as a feedback mechanism to control the Motors speed in this Lab. The ESSUR2 Robot is calibrated on the smooth flat tile surface of the Electronics Lab to set PWM value to forward speed and rotation of Robot. The Robot is then tested using different loads on Robot, uneven surfaces and surfaces with bumps, to demonstrate the limitations of the routines students wrote.

In Lab 9 students implement a PID control system is used to insure the ESSUR2 Robot maintains correct speed over rough surfaces. The Encoders are used for feedback to the PID control system. Students provide a limit feature to the ESSUR2 to contain excessive accelerations and decelerations. Again the ESSUR2 performs a straight line maneuver followed by being rotated about the center point of the front axle. The Robot is tested again using different loads, surfaces, to show the improvement the enhanced system makes to performance of the Robot.

The IMU is utilized in Lab 10 to determine the position of the ESSUR2 Robot. The students make the ESSUR2 perform maneuvers using the IMU to measure distance and rotation.

Data Fusion is the subject of Lab 11 since position can be measured using encoders and the IMU. Using Data Fusion algorithm, students combine the best aspects of both sensors to get the most accurate measurement of position and Robot orientation.

The ESSUR2 Robot is controlled using a simple ROS system in Lab 12. Students follow a test plan to bring up the ROS system and ESSUR2 Robot. A Linux Bash script with static ROS static publisher commands controls the Robot to move in a per-determined path in the Electronics Lab. Students can monitor the progress of the ESSUR2 Robot on the Simulator Screen as it moves through the Lab.

The ESSUR2 ROS System incorporates the ROS Navigation Stack. There is a Microsoft Kinect 3D Camera mounted on top of the ESSUR2 that provides point clouds to the simulator. The Microsoft Kinect 3D point cloud is converted to a laser scan topic to provide collision, and Simultaneous Localization and Mapping (SLAM).

The major change from the ESSUR to the ESSUR2 is left/right motor control, data fusion of encoders and IMU are moved from the ROS Robot Driver Node to the MBED micro-controller in the Robot Base. The Linux Bash Script sends a series of vel_cmd topic messages to the ROS Robot Driver Node to move the Robot in the Lab. The vel_cmd twist message sets the linear velocity and angular velocity of the Robot Base. The ROS Robot Driver Node publishes the odom topic giving the position of the Robot to the Navigation Stack. The ROS Robot Driver Node also sends the tf transform with the Robots position to properly display the Robot model in the ROS RVIS Robot Simulator.

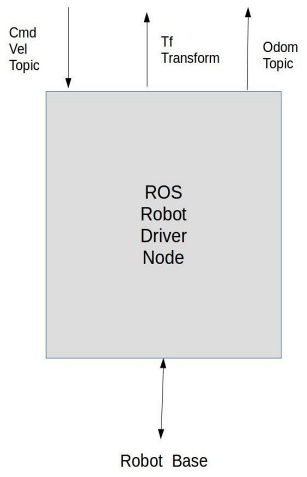
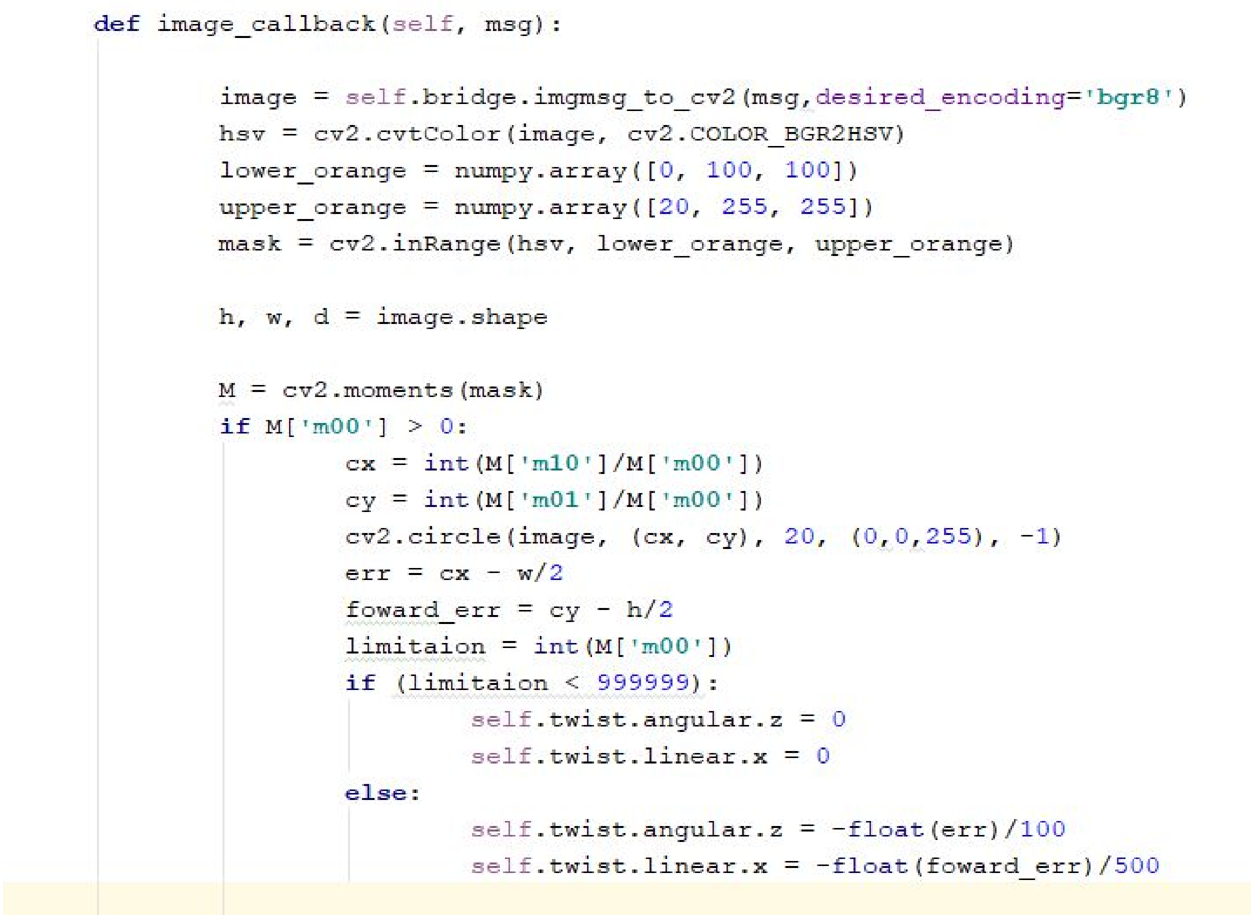


Fig 4. ROS Node

Because the ESSUR2 ROS Robot Driver Node is simpler than the ESSUR version, the node function is to reformat data to or from the serial interface to ROS topics and transforms.

In Lab 13, students will learn the basics of Open CV (computer vision algorithms). Students will be identifying objects of color and will have the drone follow such an object. The OpenCV will be downloaded and students will have to import the necessary libraries into ROS and write a simple algorithm to have the Parrot Bebop 2 drone track and follow the object using the respected algorithm. Resources such as *"Effective Robotics Programming with ROS, Third Edition"*, by Anil Mahanti, will provide students with information on how to create the necessary algorithm. Figure 5, shows part of the script that students will use to track a colored object.



```
def image_callback(self, msg):

        image = self.bridge.imgmsg_to_cv2(msg,desired_encoding='bgr8')
        hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
        lower_orange = numpy.array([0, 100, 100])
        upper_orange = numpy.array([20, 255, 255])
        mask = cv2.inRange(hsv, lower_orange, upper_orange)

        h, w, d = image.shape

        M = cv2.moments(mask)
        if M['m00'] > 0:
                cx = int(M['m10']/M['m00'])
                cy = int(M['m01']/M['m00'])
                cv2.circle(image, (cx, cy), 20, (0,0,255), -1)
                err = cx - w/2
                foward_err = cy - h/2
                limitaion = int(M['m00'])
                if (limitaion < 999999):
                        self.twist.angular.z = 0
                        self.twist.linear.x = 0
                else:
                        self.twist.angular.z = -float(err)/100
                        self.twist.linear.x = -float(foward_err)/500
```

Figure 5.0 Algorithm used to do color tracking, and having a robot respond to its location

Lie Groups and Algebras in Robotics

In robotics, movement is a key component. For robots with links, this movement is interpreted as a rigid body motion. In 2D or 3D these rigid body motions are translations and rotations, and in mathematics this is well understood. Differential Geometry has described these movements in even more complex spaces through Lie Groups and Lie Algebras, fortunate for us rigid body motion in 2D or 3D is a special case of the spaces. Here are the most important Lie groups and Lie algebras used in robotics [16] and [17]:

Lie Group	**Lie Algebra**
$SO(n) = \{R \in R^{nxn}: RxR^t = I, detR = 1\}$	$so(n) = \{R \in R^{nxn}: RxR^t = I, detR = 0\}$
$SE(n) = \left\{\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}: R\epsilon SO(n), t\epsilon R^n\right\}$	$se(n) = \left\{\begin{pmatrix} w & v \\ 0 & 0 \end{pmatrix}: w\epsilon so(n), t\epsilon R^n\right\}$
$GL(n, R) = \{A \in R^{nxn}: detA \neq 0\}$	$gl(n, R) = M_n(R)$

2 Dimensional Cases

When n=2, we have the Lie $SE(2)$ and the corresponding Lie algebra $SO(2)$, they describe the rigid body motion in two dimensions

$$SE(2) = \left\{\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} / (R, t)\, \epsilon SO(2) x R^2\right\}$$

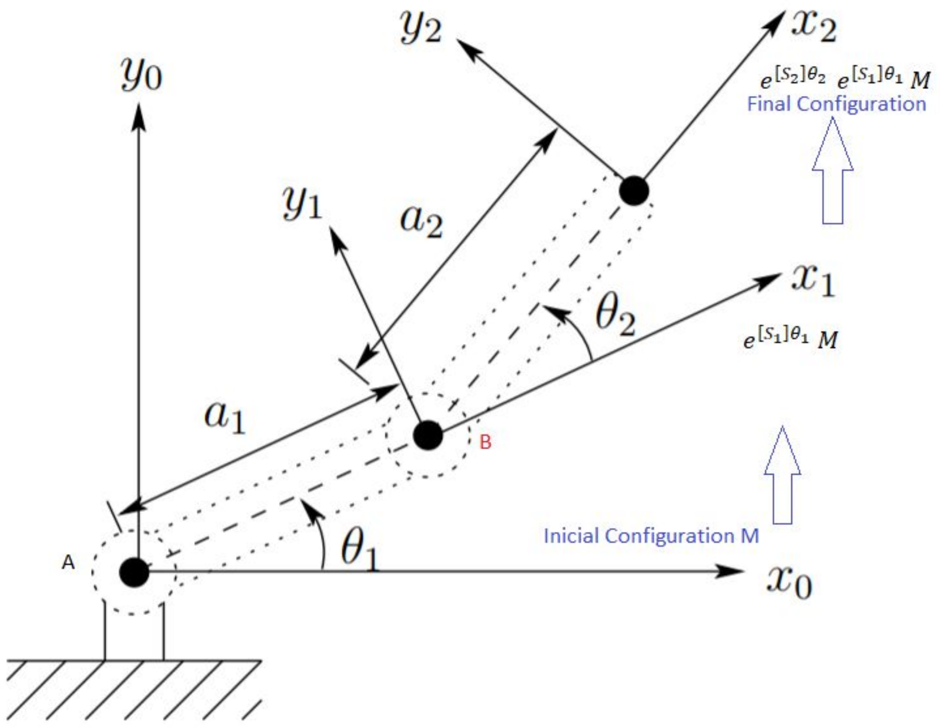
Notice and element T ϵ SE(2) is :

$$T = \begin{pmatrix} r_{11} & r_{12} & d_x \\ r_{21} & r_{22} & d_y \\ 0 & 0 & 1 \end{pmatrix}$$

Where $R = (r_{ij})$, $t = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$ and $0 = [0 \quad 0]$

Example;

Consider a two dimensional robot arm from the initial configuration M (Axis $x_o - y_o$), then it make a rotation θ_1 on the first link (A), followed by another rotation θ_2 on the second link (B)



The final configuration $T \in SE(2)$ can be represented by:

$$T = e^{[S_2]\theta_2} e^{[S_1]\theta_1} M$$

Where $[S]\theta = \begin{pmatrix} 0 & -\theta & d_x \\ \theta & 0 & d_y \\ 0 & 0 & 0 \end{pmatrix}$

$$e^{[S]\theta} = \begin{pmatrix} cos\theta & -sin\theta & m \\ sin\theta & cos\theta & n \\ 0 & 0 & 1 \end{pmatrix}$$

Conclusion

Although we learned of the V-REP simulator late in the design of the course, we felt it had excellent features for teaching Robotics. The V-REP Simulator can also be integrated with Matlab and ROS.
We designed the course with the idea that Labs would be hands-on, but also have some Labs where a simulation is used, to expand activities beyond equipment in our Electronics Lab.

References

[1] Tobe, Frank, “30+2 research reports forecast significant growth for robot industry”, Nov 21,2017, The Robot Report retrieved from https://www.therobotreport.com/302-research-reports-forecast-significant-growth-robot-industry/#

[2] Kevin M. Lynch and Frank C. Park, "Modern Robotics: Mechanics, Planning, and Control", Cambridge University Press, 2017, ISBN 9781107156302.

[3] Richard M. Murray. California Institute of Technology. Zexiang Li. Hong Kong University of Science and Technology. S. Shankar Sastry. University of California, Berkeley, “A Mathematical Introduction to. Robotic Manipulation”, cc1994, CRC Press. Retrieved from https://www.cds.caltech.edu/~murray/books/MLS/pdf/mls94-complete.pdf

[4] JM Hervé ,”The Lie group of rigid body displacements, a fundamental tool for mechanism design”,1999, Retrieved from https://www.sciencedirect.com/science/article/pii/S0094114X98000512

[5] Eade, Ethan, “Lie Groups for 2D and 3D Transformations”, Updated May 20, 2017 Retrieved from http://www.ethaneade.com/lie.pdf

[6] Asad Yousuf, Savannah State University; William Lehman, Bill's Robotic Solutions; Mohamad Mustafa, Savannah State University; Mir Hayder, Savannah State University, “Low Cost Robot Arms for the Robotic Operating System (ROS) and MoveIt”,Paper ID #14789, ASEE 123rd Annual Conference and Exposition, 2016

[7] Code for Modern Robotics Matlab Toolbox can be found at https://github.com/NxRLab/ModernRobotics

[8] V-REP documentation Retrieved from http://www.coppeliarobotics.com/helpFiles/

[9] Asad Yousuf, Savannah State University; William Lehman, Bill's Robotic Solutions; Mir Hayder, Savannah State University ; Mohamad Mustafa, Savannah State University, “Introducing Kinematics into Robotic Operating Systems”, pg. 39-47, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION | V7, N2, FALL/WINTER 2015

[10] Corke Peter, Robotics Toolbox, Retrieved from http://petercorke.com/wordpress/toolboxes/robotics-toolbox

[11] Corke, Peter (2013) “Robotics, Vision and Control” (pp. 1-6). Springer-Verlag Berlin Heidelberg
“Why teach robotics using ROS”. Retrieved January 30, 2016 from https://www.researchgate.net/publication/274267022_Why_teach_robotics_using_ROS

[12] Asad Yousuf, Savannah State University; William Lehman, Bill's Robotic Solutions; Mir Hayder, Savannah State University, “Environmental Monitoring Robot System”, Paper ID #18508, ASEE 124rd Annual Conference and Exposition, 2017

[13] Sparkfun Inc 9DoF Sensor Stick Retrieved from https://www.sparkfun.com/products/13944

[14] Vex DC Motor Retrieved from https://curriculum.vexrobotics.com/curriculum/speed-power-torque-and-dc-motors/dc-motors.html

[15] Vex Optical Shaft Encoder Retrieved January 14, 2017 from http://www.vexrobotics.com/276-2156.html

[16] Gregory S. Chirikjian . “Information Theory on Lie Groups and Mobile Robotics Applications”. 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA

[17] Vadarajan, Veeravalli Seshadri. “Lie Groups. Lie Algebras and Their Representations”. Springer-Verlag New York.

Appendix A

Topic	Description	Text Book Section and Video	Time (min)	Class #
Robots and Degrees of Freedom	Degrees of Freedom of a Rigid Body	2.1	28	1
	Degrees of Freedom of a Robot	2.2		
	Configuration Space Topology	2.3.1		
	Configuration Space Representation	2.3.2		
	Configuration Velocity Constraints	2.4		2
	Task Space and Workspace	2.5		
Representation of Rigid-Bodies in 2D and 3D Space	Rigid-Body Motions in the Plane	3.2.1 Part 1	41	3
		3.2.1 Part 2		
	Rotations and Angular Velocities	3.2.2		4
	Exponential Coordinate Representation	3.2.3 Part 1		
		3.2.3 Part 2		
	Homogeneous Transformation Matrices	3.3.1		5
	Twists	3.3.2 Part 1		
		3.3.2 Part 2		
	Exponential Coordinates Representation	3.3.3		6
	Wrenches	3.4		
Kinematic Chains and Forward Kinematics	Screw Axis in the Base Frame	4.1.1	14	7
	Examples	4.1.2		
Velocity Kinematics and Statics	Space Jacobian	5.1.1	35	8
	Body Jacobian	5.1.2		
	Statics of Open Chains	5.2		9
	Singularity Analysis	5.3		10
	Manipulability	5.4		
Inverse Kinematics	Numerical Inverse Kinematics	6.2 Part 1	14	11
		6.2 Part 2		12
Midterm				13
Dynamics	Lagrangian Formulation of Dynamics	8.1 Part 1	51	14
	Lagrangian Formulation of Dynamics	8.1 Part 2		
	Understanding the Mass Matrix	8.1.3		15
	Dynamics of a Single Rigid Body	8.2 Part 1		
	Dynamics of a Single Rigid Body	8.2 Part 2		
	Newton-Euler Inverse Dynamics	8.3		16
	Forward Dynamics of Open Chains	8.5		17
	Dynamics in the Task Space	8.6		
	Constrained Dynamics	8.7		18
	Actuation, Gearing and Friction	8.9		

Control Systems, Actuators and Sensors		No Video for:		
	Control Systems Overview	11.1		19
	Error Response	11.2		
	Motion Control with Velocity Inputs	11.3		20
	Motion Control with Torque or Force Inputs	11.4		21
	Force Control	11.5		22
	Impedance Control	11.7		23
Final				24