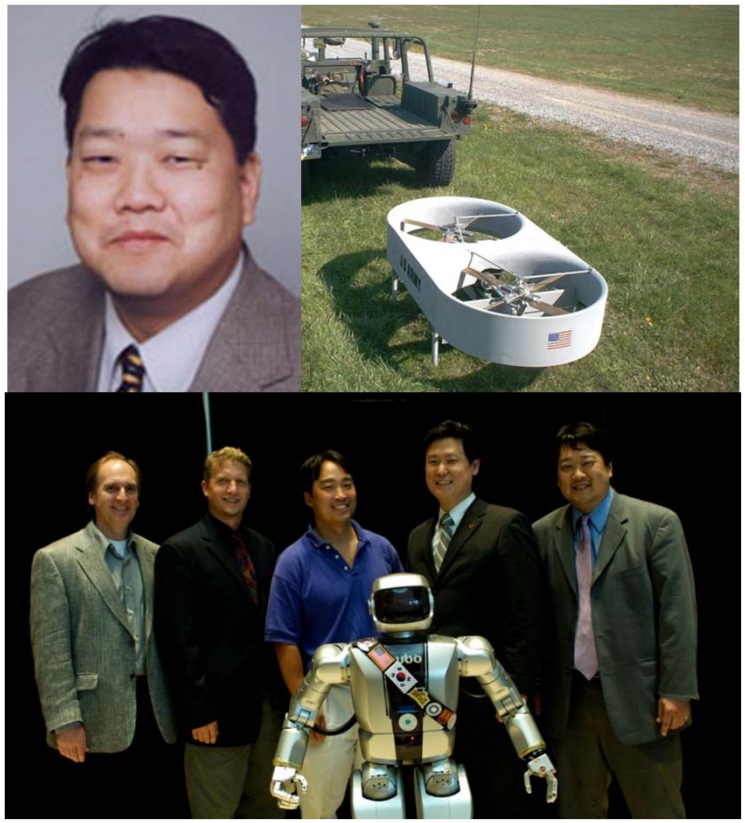
UNLV ME 729 – Robotics II (Analysis and Synthesis)

Introduction and Course Philosophy

Paul Oh: Background (Joined UNLV July 2015)



- 5+ years industry before Drexel
- ME Professor since 2000
- Advisor: ASME 10-years
- Advisor: 8+ SD teams
- 3 SD awards (COE, ASME, Baiada)
- $10M+ in research/design funds
- 140+ Journals, Books, Proceedings

- Drexel Autonomous Systems Lab
- NASA, ONR, Boeing Fellow
- NSF Robotics Program Director (2008-2010)

Enjoy creating courses where labs reinforce theory

Motivation for UNLV Robotics Course Sequence

Background: Only 2 Robotics Courses in the Catalog

ME 425/625 Robotics Catalog Description: "Instruction to basic concept and theory behind motions generated by robot manipulators; kinematics, dynamics, and trajectory generation. Design of basic feedback position controllers and computer simulation techniques of robot dynamics and control system."

To better meet needs, changes: self-contained hands-on course; instill skills in mechanisms, electronics, and programming; applicably beyond robotics e.g. automation, mechatronics, and entertainment engineering

ME 729 Advanced Robotics Catalog Description: "In-depth study of advanced automation concepts and robotic manipulators. Topics including 3-D kinematics, trajectory generation, compliance analysis, dynamic control of robotics along with concept of assembly operations and machine vision."

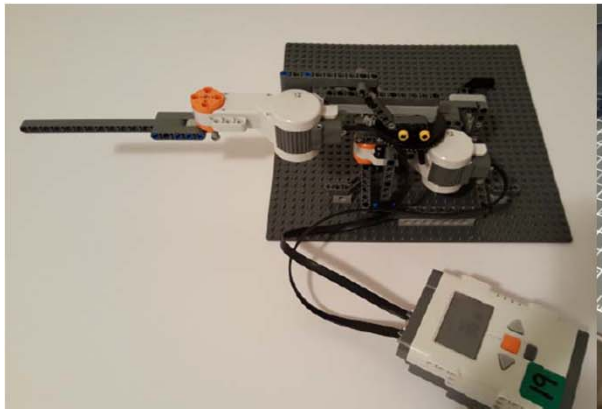
To better meet needs, changes: leverage ME 425/625 to analyze and synthesize robots; hands-on course using real hardware; motivate underlying math and study of the theoretical concepts.

My “Issues” with Robotics Courses like ME 729

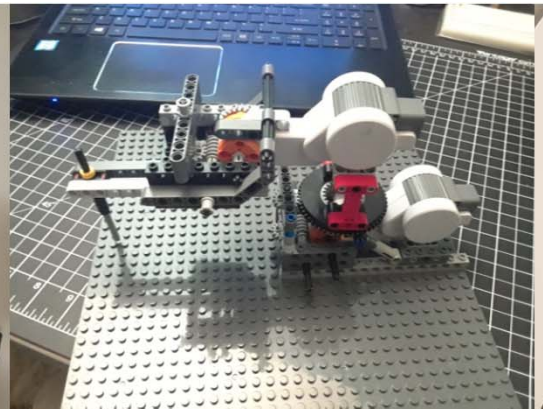
- Students fail to see “big picture” when overwhelmed by underlying math (matrix theory, set theory, linear and non-linear math, etc)
- Many offerings are purely mathematical and/or analyzed mathematically
- Simulations often used to reinforce concepts – but seldom translate into practical and real-world practice
- Software Toolboxes encapsulate the math – but often these remain “black boxes” and their limitations remain unknown to the student
- Software often demands licensing, add-on package management, and often yearly subscriptions and/or updated Operating Systems
- Software often dependent on Operating System (e.g. Windows, Linux, Mac)
- Textbooks essentially present the same material – but much is publicly and freely available online
- Hardware for hands-on labs is eclectic, lacks scalability, and/or requires maintenance

ME 729 Approach to Overcome “Issues”

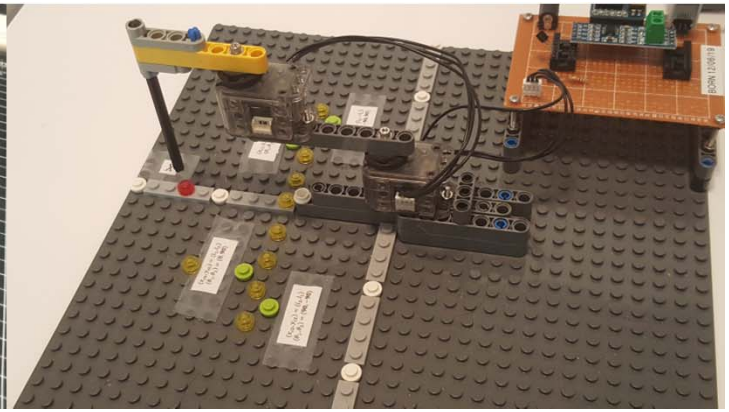
- 2-link planar manipulator is good example – its analysis and synthesis scales to higher order manipulators
- LEGO has standard spacing (e.g. 8 mm/stud), ubiquitous, scalable, affordable, and low maintenance
- LEGO Brick is self-contained and rugged with RS-485, I2C, Bluetooth, buttons, standard cables, battery and screen
- Dynamixel XL-320 smart servo has suitable footprint for LEGO, has built-in encoders, has better torque and low backlash, can daisy-chain and uses RS-485 communication
- NXC Programming: C-like syntax, free, and widely used
- Scilab: Open-source alternative to Matlab/Simulink and their libraries
- Code Blocks: Open-source GCC compiler and IDE (Windows, Mac, Linux)



Version 1 (2015-2016): Too much backlash



Version 2: Complex build



Version 3 (2019-2020): XL-320 Smart Servos

ME 729 Course Outline and Syllabus

- Course Outline (including Grading, ABET, Conduct): Linked on ME 729 home page
- Course Schedule (i.e. Syllabus): Link on ME 729 home page
- Academic Policies: in Syllabus or scan QR code below

Next Steps

- Pick up ME 729 Kit ($50 post-dated check; check cancelled when all parts returned)
- Lecture/Lab: Every Monday **18:00-20:30** (1325 E. Flamingo Rd)
- Bring Kit and laptop to every class