CONTACTS:

Dr. Rahul Mangharam  
EMBS Program, Director  
rahum@seas.upenn.edu  
Levine 279  
Office hours: 1:30 - 2:30 pm

(Liz) Wai-Ping Ng  
EMBS Program, Associate Director  
PRECISE Center, Manager  
wng@cis.upenn.edu  
Levine 313  
Cell: (267) 909-0706 [Call/Text]

WORKSPACE/COMPUTER LABS:

Most graduate students work in the Moore 100 lab.

PRECISE Center (Software Lab): Levine 315  
PRECISE Center (Hardware Lab, mLab): Levine 279  
PRECISE Center (IoT Lab): Pennovation Works, 3401 Grays Ferry Avenue  
Building 6197, Room 152

Information about all School of Engineering labs can be found @  
www.seas.upenn.edu/cets/labs/
ETHICS

As the University of Pennsylvania is an academic community, its fundamental purpose is the pursuit of knowledge. Essential to the success of this educational mission is a commitment to the principles of academic integrity. Every member of the University community is responsible for upholding the highest standards of honesty at all times.

The CIS department encourages collaboration among graduate students. However, it is important to recognize the distinction between collaboration and cheating, which is prohibited and carries serious consequences.

Cheating may be defined as using or attempting to use unauthorized assistance, material, or study aids in academic work or examinations.

Some examples of cheating are:
- collaborating on a take-home exam or homework unless explicitly allowed
- copying homework
- handing in someone else's work as your own; and plagiarism
- plagiarism also includes a student using the structure of an original program exactly while changing only a few details that do not affect the meaning of the program.

If you have a question about what constitutes cheating, ask the course instructor or contact the CIS Graduate Coordinator.

Resources/Information:

- CIS Graduate Handbook: www.cis.upenn.edu/grad/gradhandbook.shtml#Ethics
- Academic Integrity & Avoiding Plagiarism: www.cis.upenn.edu/grad/documents/1Guidelinesforusingsources-gradrev.pdf
- Code of Academic Integrity: www.upenn.edu/academicintegrity/ai_codeofacademicintegrity.html
- Academic Integrity Guidelines for Students: www.upenn.edu/academicintegrity/
- Office of Student Conduct: www.upenn.edu/osc/
- Academic Resources: www.cis.upenn.edu/grad/academic-resources.shtml
Welcome to the Graduate Program of the Department of Computer and Information Science at Penn. We look forward to working with you. Below is information you should find useful.

- **PENNCARDS/ID CARDS** - [cms.business-services.upenn.edu/penncard/](http://cms.business-services.upenn.edu/penncard/)

- **COMPUTER ACCOUNTS & RESOURCES FOR COMPUTER ASSISTANCE** – [www.seas.upenn.edu/cets/](http://www.seas.upenn.edu/cets/)

- **SECURITY** - Penn is in a big city that, like all cities, has problems with crime, which sometimes intrude upon academia, so be alert. REPORT ANYTHING SUSPICIOUS OR ANY CRIMINAL ACTIVITY TO THE DIVISION OF PUBLIC SAFETY - Penn Police, [www.publicsafety.upenn.edu](http://www.publicsafety.upenn.edu), at 511 (on campus) or (215) 573-3333 (off campus) the Philadelphia Police Department can be contacted at 911.

Penn’s Walking Escort Services, Transit Ride Service and other safety information: [www.publicsafety.upenn.edu/safetytips/](http://www.publicsafety.upenn.edu/safetytips/)

The Penn Shuttle Bus service – [cms.business-services.upenn.edu/transportation/types-of-services/penn-shuttles.html](http://cms.business-services.upenn.edu/transportation/types-of-services/penn-shuttles.html) - provides transportation to Center City, University City, & West Philadelphia.

- **ACADEMIC RESOURCES (REFERENCE LINKS)** - [https://precise.seas.upenn.edu/education/masters-program](https://precise.seas.upenn.edu/education/masters-program)

- **GENERAL RESOURCES, INCLUDING LINKS TO THE PENN GRADUATE STUDENT CENTER, HOUSING, STUDENT HEALTH, & COUNSELING & PSYCHOLOGICAL SERVICES, FINANCIAL SERVICES, STUDENT GROUPS, ETC.** - [www.cis.upenn.edu/grad/general-resources.shtml](http://www.cis.upenn.edu/grad/general-resources.shtml)
  If you have any question or concern please contact Liz, [wng@cis.upenn.edu](mailto:wng@cis.upenn.edu)

- **MAILFOLDERS** - Each graduate student has a mailfolder on the 3rd floor of Levine, opposite Levine 302. As the mailfolders are not secure, it is not a good idea to use them to receive personal mail. If you are expecting an important mail, you can put “[Your full name], Attn: Liz W. P. Ng, 3330 Walnut Street, Levine Hall Room 302, Philadelphia, PA 19104.”

- **HOPE** - We sincerely hope that you find your graduate career with us to be enjoyable, exciting, and rewarding.
EMBS Course Suggestions – Informal and un-edited advice from EMBS Alumni
(Last updated: August 2017)

Based on alumni specializations and interests, we asked them to identify which electives they found most useful/interesting. This is informal advice (use it as your own risk) and un-edited (so excuse the grammar😊)

Robotics

- MEAM 520: Intro to robotics - Teaches introduction to coordinate transforms and simple control theory. Very useful if you plan on taking ESE 650 and MEAM620
- CIS 520: Machine Learning - Gives a good mathematical background to a lot of algorithms and lays the foundation for Learning in robotics. Less hands-on and less programming than CIS519.
- MEAM 620: Advanced robotics - Hands-on course focused on control and planning algorithms. Heavy course load. Has a pre-req of MEAM520
- CIS 580: Machine Perception - Vision algorithms using geometric techniques, involves concepts such as object recognition, deriving 3d images from multiple 2d images, etc.
- ESE 650: Learning in robotics - Project intensive course with a heavy course. Projects involve implementing a number of machine learning algorithms in robotics on datasets from robots of DARPA challenges.

Machine Learning

- STAT 542: Bayesian Methods & Computation (Wharton Statistics Course) - for building up knowledge if you plan on taking CIS520
- CIS 520: Machine Learning - Gives a good mathematical background on how algorithms work and choosing algorithms for applications.
- CIS 521: Artificial Intelligence - A theory intensive course focusing on AI algorithms with some programming in python.
- CIS 519: Introduction to Machine Learning
- CIS 580: Machine Perception
- CIS 680: Vision & Learning
- ESE 539: Neural Networks: Theory and Applications

Machine learning alone is not enough, it just gives students an idea of how machine learning works (CIS 519/520), computer vision (CIS 580/581) is a very important application field of machine learning, Deep learning (CIS 680) is a must for AI career seekers. Even so, to get in this field, one must obtain a doctoral degree. All open positions required a PhD.

IoT and Core Embedded Systems

- ESE 519: Real-Time Embedded systems
- CIS 573: Software Engineering - android app development
- CIS 505: Software Systems - important distributed system algorithms and techniques
- CIS 553: Networked Systems - Basics of networking
- Fun elective - IPD course on smart objects (IPD 521/528) - Good experience of working with students from design school who are not actually engineers.
• Fun elective - ESE 527: Design of Smart Systems - Relatively new course on designing smart systems in a principled manner. Apply data-driven methodology to design and simulate a smart system prototype.

**Networking**

• ESE 519: Real-Time Embedded systems - hands on bare metal programming  
• CIS 505: Software Systems - hands on network programming  
• CIS 700: Operating Systems - hands on designing an operating system

**EMBS alumni’s comments for courses:**

• ESE 519: RTES - For people looking opportunities in firmware domain, a must take in first semester for internships.  
• CIS 501 : Computer Architecture - architecture is a foundation that everyone should take, and it applies well to both the core embedded topics (cycles per instruction performance and cache coherence) and distributed systems (memory hierarchies and parallelizing operations). Professor Joe D. focuses on how caching improves performance in avoiding high-latency memory accesses and adding levels of indirection to increase functionality, and you will see applications of those two ideas in every single class you take.  
• CIS 505 : Software systems - distributed systems and socket programming taught in this course, applied to embedded software in robotics and connected embedded systems. Hence should be taken in 1st or 2nd semester.  
• CIS 550: a really important complemet to CIS 505 (software systems). Since software systems focused on the distributed applications/algorithms and communication between systems, but assumed reliable, distributed storage. CIS 550 teaches how data is persisted, especially in modern, distributed web systems. Most often, engineers take for granted how this works, especially in projects people end up doing on cloud infrastructure like AWS. Most often, students didn't understand how or why databases are needed to persist and share information between asynchronously executing programs (either on the same machine, or remotely).  
• CIS 573 : Software Engineering - Introduces agile, android and unit test cases which might prove to be helpful for students attempting CS based job interviews. Otherwise, this course is not much useful for students interested in embedded or robotics domain.

"CIS 573 is not a difficult class, but it teaches you how to think about writing high quality code. In my first few weeks at work, I've been working with the firmware team to start adding unit tests (and mocking certain functionality so it's testable) and redundant code like retry blocks to make our embedded systems more robust. No other course touches these topics, and it covers a high-level, but pretty useful review of data structures and algorithms performance that comes up in interviews. The project is a big part of the course, but I felt like most people chose projects that weren't a great fit for our program because most of them are Android applications. It was definitely an easier path to choose a predefined project, but it's also basically open-ended and I think people could do a better job proposing projects that intersect with embedded systems in interesting ways, like adding mobile interaction or networking to standalone systems, or on development and design tools."
• MEAM 520: Intro to Robotics - A good course to have in your first semester if someone is planning to get into robotics or advanced robotics in future and is not familiar with kinematics and transformations.
• CIS 553: Networked Systems - A good course for students interested in networks, embedded systems, or IOT.
• CIS 520 Machine learning vs CIS 519 Intro to Machine Learning: This is the most common question students asked. CIS 520 is deeper and mathematical, suitable for students who want to take some other advanced courses in machine learning in upcoming semesters. CIS 519 is a bit more practical orientated using python and can be targeted by those who want to know what machine learning is for the sake of knowledge and are not sure if they will pursue advanced courses.
• ESE 650: Learning in robotics - A must take for anyone interested in machine learning and robotics. A good perspective in robotics is essential. People just interested in ML and not in robotics might get completely lost.
• CIS 620: Advanced AI - For anyone who wants to advance in ML but not much fan of robotics.
• CIS 581: Computer Vision - give students hands-on experience with CV and bare metal algorithms behind it.
• CIS 555: Internet & Web systems - great reviews for this course. Heavy course load. Makes the student a pro in Java. Should be taken in 4th semester if you are interested in web and want to get a hands-on feel for developing a search engine. Should take it in 2nd semester if you are looking for internship in java or web based software opportunities. For students looking at Deep Learning, Hadoop experience is a requirement and this course covers the same.
• CIS 700: OS Design and Implementation - cover 80% of Apple's job interview questions.
• MEAM 513 - Feedback control design - worth taking for anyone looking deeper into robotics. Spring semester.
• Mean 620 - Advanced Robotics - Control and planning algorithms explained with examples of quadcopters. A good pair with or a step up after MEAM 513. Might as well directly take it and skip MEAM 513

Additional tips from EMBS alumni:

• Sit in other courses (to learn more) you did not register
• Take moderately hard classes to get good grades during the first semester. Challenging yourself is great but one C grade will affect the overall GPA.
• Fortune 500 companies screen resumes with GPA. Easier to get interview calls if you have a GPA of 3.5+.
• Try to have a combo of RTES and SS (in the first semester) and OS (in the second semester) so that you enhance your programming skills and pass your programming interviews.
• Companies tend to pick candidates who have worked in open-ended projects (things not built before, and you are not sure if you will succeed).
• Knowledge from CIS 505 has proven to be helpful for any internship. A good understanding of socket programming in C is extremely useful across the board.
• CIS 550 (database & info systems) and CIS 573 (software engineering) are practical courses.
• CIS 541 - The "formal" part is very important for industry. Tesla interview asked a lot of questions on RTOS (explain different types of scheduling, and priority of scheduling, how to implement RTOS on a micro-controller, how is the load split across different cores. Questions around code conversion from state machine (process). Need to know deeper knowledge of what is happening inside the microcontroller.
• Be proactive to ask professors for feedback - so you can fix any issues if they exist instead of causing compounding errors from one part to the next.

• Recommend CIS 555, CIS 505, CIS 501 - Interviewers asked C/C++ Qs, memory mgmt, multithreading knowledge, algorithm (high level), assembly programming, cache/virtual memory (computer architecture), operating systems (covered in CIS 505 + CIS 541). Also asked "how will you deploy the code, what kind of different texts you will use, what kind of benchmark you will implement - software development life cycle", "design the whole hardware pipeline" and "software interface".

• Try to take CIS 540 and an operating systems class (or ESE 519) before taking CIS 541.

    From our alumni: "The material in 541 assumes real depth in a lot of areas, and I think the project is really interesting but most people don't get a chance to do it well because they are learning topics for the first time that are assumed to be well-known."

• If you have at least two years of working experience, and your boss is thinking to promote you as a system engineer, then you need to have a strong base in system engineering perspective. CIS540 builds your theoretical knowledge related to embedded systems and CIS541 is a complimentary course to CIS540, which provides you with practical knowledge.

    **But keep in mind to take CIS540 before CIS541.**
Sample Schedule

Note: CIS 501 will not be offered Fall 2018. A good suggestion is to swap it with CIS 505, and take CIS 501 in Fall 2019.

Year 1 - Fall
- CIS 501 - Computer Architecture
- ESE 519 - Real-Time Embedded Systems
- Elective

Year 1 - Spring
- CIS 505: Software Systems
- CIS 540: Principles of Embedded Computation
- Elective

Year 2 - Fall
- CIS 541: Embedded Software for Life-critical Applications
- Elective
- Elective or Thesis/Independent Study

Year 2 - Spring
- Elective or Thesis/Independent Study

Recommended Electives based on Career Track

Robotics
- MEAM 510: Mechatronics
- MEAM 520: Intro to Robotics
- MEAM 620: Advanced Robotics
- CIS 520: Machine Learning
- CIS 580: Machine Perception
- ESE 650: Learning in Robotics

Machine Learning (Note: All open full-time positions required a PhD)
- ESE 539: Neural Networks: Theory and Applications
- STAT 542: Bayesian Methods & Computation
- CIS 519: Introduction to Machine Learning or CIS 520: Machine Learning (Theoretical)
- CIS 521: Artificial Intelligence
- CIS 580: Machine Perception
- CIS 680: Vision & Learning

IoT
- ESE 516: IoT Edge Computing
- CIS 548: Operating Systems
- CIS 553: Networked Systems
- CIS 573: Software Engineering