METR 4202 -- Advanced Controls & Robotics

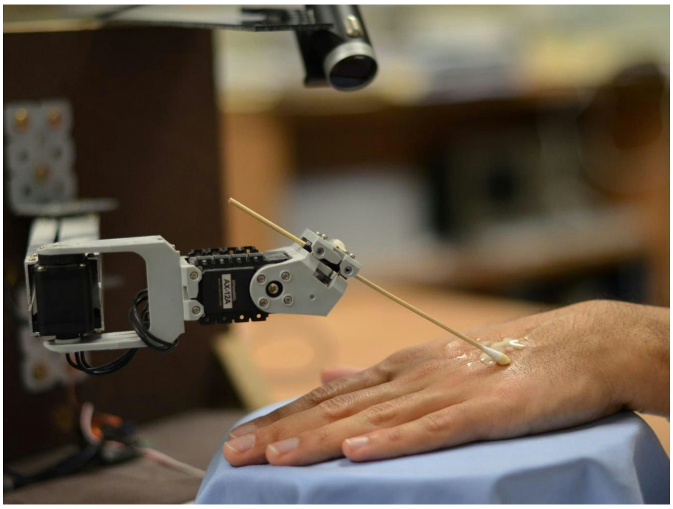
-- Lab 3: Systems & Control --
Robotics: Helping Hands$_\beta$

Puras deus, non plenas adspicit manus.
(God looks at pure, not full, hands.)
Publilius Syrus

Objective

It's been said that the goal of this class is to understand robotics like the back of one's hand. This laboratory adopts this view and considers the principles of kinematics, dynamics, sensing, trajectory planning and motion control aspects of a robotic system that can not only give an helping hand, and perhaps help hands.

This laboratory explores robotics hands-on (on hands!). Motivated by the problem of applying creams and ointments, this project looks at the problem of placement and coverage in dynamic cases. As with many things, it can be challenging at the pointy end. Notably, in this case, skin deflects in addition to general hand and arm motion. Thus, the lab is structured in tiers, from tracking, to tracing, to touching, to helping, to ink placement (this is used as a proxy for ointment, because we wouldn't want the robot or hand to get creamed).



Requirement

Design and build an automatic robotic system to place an ink dot, adhesive bandage, or pattern at a prescribed location on the back of an arbitrary adult hand of either handedness. Depending on the task, the location of the point on the hand may be specified as a metric location relative to a base frame or specified relative to an anatomical feature, such as the base of the thumb (or in an anatomic sense, the location where the proximal phalanx connects to the metacarpal).

The final system should be able to:

- Start with hand(s) and potentially clutter in the workspace
- Finish with the appropriate mark(s) on or around the hand as per the candidate task

Importantly, given that the laboratory is about being handy, the system should do this completely and strictly within ten (10) minutes. The system must operate with four (4) robot actuators or less (preferably three).

Workspace and Scene Structure

The scene is similar to Laboratories 1 and 2, but with the extension of much greater depth so as to allow an adult to move their and hands and arms comfortably. As with the previous laboratory, there is a basic level of operation and advanced level of the workspace. At more exceptional levels may include clutter in the form of dry-erase pens (of the standard and colours typically found on campus, such as the Nikko 5000), dry-erase erasers foreign currency, keys, soda beverage cans, small chocolates, coins, paper cups and other small random items (max height of 10 cm). The workspace will be defined with varying levels of structure and clutter, with lower performance standards having more structure. This is outlined as follows:

Item	Basic Level	Advanced Level
Background	Monocolor non-white paper (teams may remove it)	(Anything)
Clutter	No	Yes
Obstacles	No	Yes
Colour Calibration Target	Maybe	No
Central frame placement	Static (Optionally team-placed)	Dynamic (Random)
Calibration Pattern	Maybe	No (except for one initial use)
Motion Planning Expected	Maybe	Yes
Hand motion	Mostly static	Maybe dynamic
Hand/Scene Geometry	Mostly 2D (e.g., flat hand)	2D and 3D (hand maybe posed)
Hand prescription (location maybe set in advance)	Allowed	No

System Candidate Tasks

The focus of the laboratory is advanced control and task planning along with integrated sensing and robot design. To demonstrate robotics and control operation, five tasks are proposed:
These include:

1. **Outline:** Draw an outline of the person's hand on a sheet of paper underneath their hand in a manner that may go automatically from a given start and stop point in a way that avoids obstacles.
2. **Acupuncture:** Be able to touch the hand at specific metric points. Operation should avoid obstacles on the hand or fingers (such as rings, bangles, sleeves, wristbands or watches).
3. **Band-Aid:** Be able to place a standard size (~19 × 72 mm) plaster bandage such that the woundpad covers key spots of interest. (The sterile paper and backings may be assumed to have been removed)
4. **Line:** Be able to draw a specified line or shape outline on the back of the hand
5. **Cure-all:** Be able to draw and fill in a region with full coverage.

As with Laboratory 2, the tasks receive greater points (or in the colloquialism of this laboratory more claps) for greater accuracy and for operation in more advanced level scenes. There is more credit for operation in cluttered environments (i.e., planning around obstacles) and for robustness.

Unlike previous laboratories, the tasks, while independent, are tiered. That is, the latter, higher level (and more difficult) tasks attract more credit. Teams need not be able to try or complete all candidate tasks. Also, if teams which to attempt multiple tasks they must do so in a manner such that the tasks performed are contiguous (e.g., tasks 1-2-3, tasks 3-4, tasks 3-4-5, etc.). Random or disjointed task orders (e.g., tasks 1-3-5) will not be recognized.

For this laboratory, operation in advanced level scenes results in more value (ovation). Partial credit is determined via the following (general) scale:

- **Attempt:** The system operates, but with significant error (>90%): **½ clap**
- **Some operation:** The system makes at least one successful mark/task: **1 clap**
- **Applause**: The system operates, but has errors: **2-3 claps**
- **Ovation: operation:** The system operates fully with little (<20%) to no errors: **4-5 claps**

Other Robotic Systems, Languages and Cameras

Teams seeking to do advanced level operations may experiment with (at their own risk) next year's proposed METR4202 robot hardware kit (based on three Dynamixel AX-12A actuators). Kits are available to teams on a limited, first in, best dressed basis.

Teams may elect to use programming languages and systems other than Matlab, such as Visual C++ or Python (i.e., the class is language / system neutral). In particular, teams may choose to use OpenCV. Teams may choose to operate the Kinect's RGB camera in high resolution mode. Similarly, teams may use another (web) camera on the proviso that the camera is autonomous (i.e., it operates **without manual intervention**).

Marking: Assessment Criteria for Overall Lab Mark

When pointing fingers, remember that three point towards oneself. That notwithstanding, let us consider the process of handing over grades:

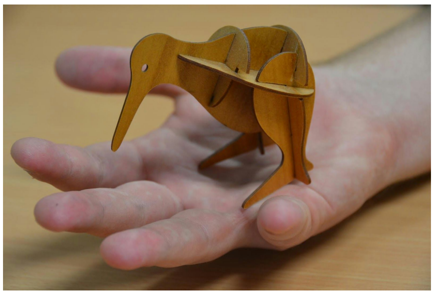
In short for the following grade levels:

- 1-3: Teams attempt and are somewhat successful at 1 or 2 of the system tasks in basic environments.
- 4: Teams attempt 1 to 2 system tasks and are marginally successful at these at least in basic environments.
- 5: Teams attempt 2 to 3 system tasks, with some success in an advanced environment and are robustly successful at all core tasks even, including the presence of noise/clutter/etc.

- 6: Teams attempt ≥2 tasks in advanced environments and are successful at tasks 2 or higher.
- 7: Teams attempt ≥3 tasks and are robustly successful at either tasks 4 or 5.

As a rough guide that mapping between claps and the grades is:

Grade	Applause Level	Description
2 (20-45)	2 claps	At least one task performed.
3 (45-50)	3-6 claps	Very substandard performance,
4 (50-65)	7-10 claps	**Basic level operation.** For example, you are able to operation with satisfactory accuracy.
5 (65-75)	11-15+ claps	**Intermediate operation level.** For example, you are able to place items/lines with good accuracy.
6 (75-85)	12+ claps + ≥1 Successful Challenge	**Very good intermediate to Advanced Level performance.** For example, you are able to place points/lines with great accuracy.
7 (85-100+)	15+ claps + ≥2 Successful Challenges	**Excellent performance**. Most of the tasks are attempted well. Teams are able to robustly place lines and/or cover areas in complex scenes with superb accuracy.



External Sites/Programs

Some external programs and site that might help with the process are:

- **CONDENSATION** -- (M. Isard and A. Blake. "Condensation—conditional density propagation for visual tracking." *Int. J. of Computer Vision* **29(1)**:5-28, 1998)
- **CLAMS -- Calibrating, localizing, and mapping, simultaneously**

Teams and Groups

The project will be conducted in **teams of four** (up to five maximum) -- preferably taken from within your group from Laboratory 1. You may also choose an individual from another group (or laboratory session) as long as you understand that you may not be able to work together for the final project that will draw upon work completed in this one. A requirement to pass this Laboratory is that you must be in a team and that team must be registered in Platypus[2] by October 15, 2015.

Due Date

The laboratory must be completed by **Thursday, October 29, 2015**. The code should submitted online via a source version control system (e.g., EAIT GIT, GitHub, GitLab, Bitbucket) by 11:59pm on October 31, 2015. A short team report should by 11:59pm on October 31, 2015 via the Platypus[2] submission system. Early submission is encouraged.

Demonstration

As with laboratory one, the system will need to be demonstrated. As with Laboratory 2, there will be signup times on Thursday, October 29 and Friday (evening) October 31 (i.e., during teaching week 13). During the demonstration period, teams are requested to demonstrate operation in order of the the focus area tasks in **any order** they choose. For each of the tasks, teams may repeat a task once if they choose; however, the team receives the value from either not both (i.e., repeat task demonstrations do not add).

Given the number of teams, the demonstration times (of 15 minutes total including setup, leaving 5 minutes for discussion) will be strictly enforced. It is recommended that teams come 15-30 minutes in advance of their demonstration appointment. It is also recommended that teams practise their demonstrations as time limits will be enforced even if teams have not been able to demonstrate their solutions to the five tasks (i.e., teams will receive grades not on the solutions they demonstrate not the solutions they might have, but did not deliver).

Judges

The course coordinator, lecturers and tutors will act as judges. The course coordinator will act as chief judge. All decisions made by the judges will be final, and no correspondence will be entered into. Contestants may approach the organiser about possible designs that may be questionable under the rules listed above. Any queries will be treated with the utmost confidentiality and will not be divulged.

Custom Candidate Tasks

As custom candidate tasks ideas are sent in and approved, they will be posted here for the benefit of other teams. Some approved custom advanced level and extra credit ideas are:

- TBA

Caveats

Some general "reasonable person" rules apply to the code and its execution:

- It is expected that teams will use source/version control
- Codes with fixed (predetermined) estimates are not valid (even if the value is correct).
- The use of the Matlab Camera Calibrator App may be used. However, teams will still be responsible for being able to explain how the calibration process works, particularly intrinsics, extrinsics, the minimum number of frames and points required (see also § 2.1.5 (pp. 45-49) of Szeliski, *Computer Vision: Algorithms and Applications*, 2010).
- Internet access may or may not be present -- the code should assume that it will not have Internet access during execution and thus operate in a self-contained manner. This proviso excludes UQ license servers that may be needed by the program (e.g., Matlab). A "Mechanical Turk" or "phone home" solution is explicitly disallowed.
- Memory space may or may not be cleared between challenges and submissions -- The memory space might be cleared before each function. Thus, if your routines rely on parameters to be exchanged, it should do so by writing to a file. Similarly, if certain variables names (e.g., counters) are used between functions, then be sure to initialize them correctly.
- Each team's submitted functions will be run in their own directory -- Reading other teams' files or memory is disallowed.
- All source code(s) may be assessed -- Thus, it is requested that it is commented. If custom precompiled codes are used (e.g., mex files), the source code and compilation instructions (e.g., makefiles) should also be submitted.
- Computational and memory resources -- the functions should be able to operate reasonably on a "standard" Laptop/Workstation class computer (such as the UQ EAIT PC Workstations). Judges may terminate execution after 2 minutes.

METR 4202: Always at Hand!