

Choreographic and Somatic Practices Toward the Development of Expressive Robotic Systems

contribution from Alli Nilles

One tool often used for controlling robotic systems is the “Robot Operating System” (ROS). As an undergraduate student of mine put it, “ROS is software with the sole purpose of forcing the user to open up many windows. It also has robot applications.” The user experience with ROS is notoriously bad. People trying to learn the software for the first time often quit because of how frustrating and tiresome the process is.

When I use ROS, to test robot motion strategies and controllers, there are many points in the process where I have to stop and switch my mode of interaction with the system.

First, I may use a text editor to edit code. Then I use another window to stop the currently running program, reload the code, wait for initialization, and click to another window to view the results on the simulator. I may have several different “nodes” running, for example, one for collecting camera data and one for controlling a robot arm. Each of these requires its own terminal window for editing and monitoring. For an example, see Figure 1, from an tutorial called “An Introduction to Robot Operating System,” aimed at absolute ROS beginners. Several factors here are intimidating: the requirement to use the command line, the difficulty of keeping track of all the windows, and the constant mode switching between writing code and manually managing the programs.

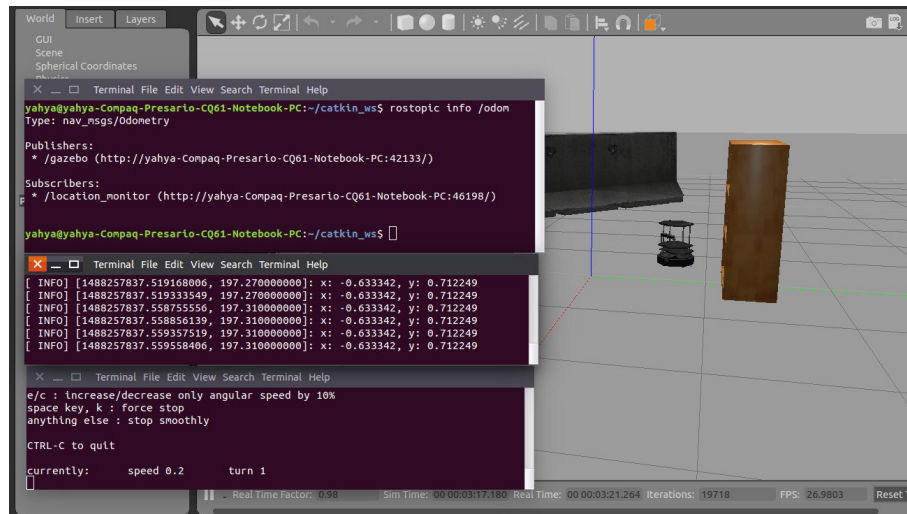


Figure 1: From <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-robot-operating-system-ros/>

Working with people in the RAD lab, such as choreographer-in-residence Catie Cuan, drove home the need to streamline this software. Artists are rarely willing to spend a week, or more, learning the technicalities necessary to use ROS. Indeed, even many technically trained people are unwilling to do so. The limitations of the software are influencing how people design projects, and limiting scientific exploration.

Developers of software for robotics are often people who do not have a clear picture of the skills and needs of non-super-users. Of course, they do an immense amount of helpful work on the software: thanks to the developers, we do not have to write our own network protocols, or hardware calibration routines. Open source software is a continual work-in-progress.

Interdisciplinary collaboration pulls us out of our bubble of experience. Skills I have learned as a computer scientist,

such as writing wrapper scripts, parsing and generating low-level code, and reading through pages of program output to debug errors, become second nature and I forget how long it took me to learn. If even PhD students in Computer Science and Mechanical Engineering struggle with the software in robotics, imagine the barriers for artists, educators, or young people.

To reduce these barriers, I started a project in Amy's graduate level class, *Movement Representation and High-level Robotic Control*, which aims to create a live-coding interface for ROS. Live-coding is a performing art especially prevalent in computer music, where many programming languages and interfaces exist for creating improvised music using sound samples and a computer. Similarly, we are creating a domain-specific language (DSL) which is immediately compiled to a process which publishes ROS messages to a robot or simulator. Thus, as you edit code, you immediately see changes on the robot with one click of a button or keybinding.

Apart from minimizing human suffering, why decrease barriers to learning, and decrease compile time while working? The goal is **flow**: the mental state of complete absorption in an activity¹. When engaged in some creative process - coding, dancing, writing, anything - the best work is done when the creator is focused, feels agency over the tools they are using, gets so wrapped up the creation that they lose track of time. If we draw an analogy: the human as choreographer, robot as dancer, imagine how difficult it would be to work with a dancer who randomly freezes, refuses to listen to commands, who only speaks one specialized language that takes weeks of dedicated practice to learn. Obviously, any human endeavour involving robots would be much more productive if the creative process was able to flow more smoothly.

In the RAD Lab, Dr. LaViers encourages her students to take dance classes. This semester, I am taking a class in Contact Improvisation. As its name suggests, this form of dance is improvised, sometimes to music, and emphasizes contact (with other humans, walls, floors, etc). I have very little experience with improvisation, with dance, music, or any other medium. But since I embarked on a project to create an immediate-feedback tool for creating robot movement sequences, I thought I should get more hands-on experience with what it feels like to spontaneously create movement.

The class has been much more impactful on how I think about movement than I could have imagined. One important impact has been an appreciation for the previously mentioned *flow* state. As most people have experienced, improvising dance is difficult. It does not feel natural at first. But with practice, it becomes easier and easier to slip into a mental state where there is no self-consciousness and the dance is done as an instinctive reaction to music, or previous movement, or the movements of other people around you. Can a robot ever be in such a state? Undoubtedly, it takes much calculation by my nervous system to make dance feel effortless. Can such natural, immediate responses be programmed into a computer? Do we have enough processing power and efficient enough controllers? Or is this phenomenon only emergent, a result of learned spatiotemporal patterns over an analog nervous system?

Either way, experiencing this creative flow while improvising movement has underscored my desire to decrease the "compile time" when working with tools for programming robot motion. Human attention is a precious resource, easily lost, and our current interfaces are making life harder for anyone who wants to work with robots - whether that's graduate students, or artists incorporating robots into a piece, or students considering whether to study robotics.

Another important role that a somatic practice has had on my perspective as a roboticist is to give me a deep respect for how complicated the body is. Sometimes, admittedly, this is discouraging - robots are an untrained child on the recorder, compared to the symphony of the human body! How can something so one dimensional as a programming language express the complexities of moving through space? But learning about movement notations and taxonomies, mainly Laban-Bartenieff Movement Studies, is a major source of encouragement and inspiration. Through this study, I am learning to precisely observe and describe movements, the same way my physics and computer science education has taught me to precisely observe and describe the physical world and computation.

During Catie Cuan's visit, we identified different choreographic "technologies." These technologies are tools that choreographers use to analyze and create movement sequences. These include notation, motif, and video recording, as well as creative tools such as arrangement of movements in space and time. Movements can be reversed, retrograded, or reflected across different axes. Similar movements can be performed at different extents or levels, or at different tempos, or with sharper or more gradual speed changes. Moreover, dancers and choreographers have identified heuristics that explain how these different technologies tend to affect how audiences interpret dances: why we tend to see some dances as frantic, others as peaceful, and what techniques create tension or capture attention. By

¹ cite some definition

automating these heuristics, it may someday be possible to program a robot at a much higher level than is currently done: the instructions from the human to the robot will be more similar to those of a choreographer. We are far from this point today, but by collaborating with artists and movement observation experts, we can accelerate progress toward this goal.