Formal Modeling of Robot Parts for Controller Synthesis

Myles Cai

October 2, 2017

1 Introduction

Robots that can be formed from the combination of various individual robotic parts have already demonstrated their potential usefulness through their versatility, flexibility, and robustness. However, a major barrier for the spread of their usage lies in the difficulty of designing a controller that can guarantee correctness when faced with complex tasks, though an increased amount of research has gone into doing exactly this with formal languages. This research project will continue building on this work, in the hopes that the new controller synthesis algorithm will be able to be applied to robotic components of all geometries and configurations.

1.1 Background

As it stands currently the lack of ubiquity of robots custom-designed from individual parts (of which modular robotics is a subset) in real-world applications has long been a symptom of the difficulty in creating a consistently convergent controller for them. Though a single challenge among many others such as hardware in the development of this field of robotics, controller design has proven to be a very active field in terms of progress. Research has been done in designing automatic controllers form a library of robot behaviors for use with modular robots in which all modules were identical [2] [5]. However, no work has yet been done in automatic controller synthesis for use with a larger platform of robotic parts to be defined by their joints, attachment points, and geometries.

1.2 Literature Review

The benefits of formal methods have been demonstrated time and time again, and its usage in various fields has been growing. Formal methods are languages, techniques, and tools for specifying and verifying systems of increasing complexity and scale, and the result of their application has been lower costs and higher quality [3]. Their efficacy at system specification, the formal requirement of what a program is to do, is especially applicable to this undertaking as it will help find convergent controllers [4].

Two papers stand out in the history of this project. One focuses on the initial development of control for modular robots, bringing to the table the demonstration that certain tasks can be translated into provably correct control for modular robots [2]. The other builds upon that work, broadening the notion of behavior properties (previously defined as traits) to include environmental properties, introducing performance enhancement by basing actions on concrete configurations, and expanding the design library [5]. The former paves the way for the latter, bringing a lot of applicable theory to the table. These two papers lead into the current research project, which will further bolster the work in this direction, by incorporating an even larger set of robot components into the group for which possible automatic controller synthesis is possible. This would be another step in enlarging the complexity of available configurations and designs for physical robot creation, as well as encouraging the possibility of more expressive task specifications [1].

1.3 Significance

As mentioned above, the successful development of a method for error-free controller synthesis for robot parts of varying properties using formal modeling would be a large step toward creating a robust and perhaps automatic controller synthesis method for an increasing number of robots. This would allow robots to handle both a wider set of expressive tasks, and tasks of greater complexity. Subsequently, this would increase the relevancy of robotics in real-world applications and as they have been doing for years, simplify and ease the lives of many.

Furthermore, this controller synthesis method for robotic parts using formal modeling would be one step forward in the creation of a more universal controller generator that could ideally be applied to a greater set of robotics.

Further work in this area would be the development of reconfigurable robot controllers, the expansion of libraries for robot behaviors, and in general, the reduction of errors in controller synthesis for all systems.

2 Research Methods

For this project we will approach controller synthesis for various robot parts with formal modeling, denoting certain geometries, actuation styles, attachment points, and other properties of these parts with a specific mathematical notation to avoid unnecessary errors and other potential pitfalls. Using a hyper-graph where edges can join multiple vertices, all properties for these components can be characterized using graph-based representations; for example physical modules would be nodes in that graph, with each of these nodes annotated with certain geometric info as mentioned above. Some possible traits that could be used to describe geometry include size, number of planes, and concavity or convexity at joints. This should be enough to successfully describe a leg, a wheel, or any other limb. Then joints or attachment points would be edges on that graph, and the actuation of them would another annotation by a command which could be completed by any kind of module. Each component will have its own behavior types, and when the robot has all its parts' behaviors defined, then the global behavior of the robot will likewise be defined.

The initial focus of this project is on navigation and path-planning tasks for the robots. This appears to be the most familiar application, as well as one of the most fundamental of robot control. We will validate our formal modeling of controllers for such through physical prototypes that we will fabricate ourselves. C++ will be the language used to construct the graph representations discussed above, and we have the resources available to design and manufacture our own robot components for assembly and testing of our hypothesis on physical models. After navigation, other possible directions that could be taken by an effective formally-modeled controller are including sensors inputs (which could be done by annotating edges with an output) or broadening the set of tasks beyond path-planning.

2.1 Personal Progress

Thus far, my time has been put into researching previous works and reading relevant papers such as those cited above. In addition, I have been working on becoming more familiar with C++, a language I have not used before, and which will be the primary language that is used in this project.

3 Timeline

Date	Event
April	Graph representation of robot parts and controller
May	Code implementation of representation, integrate existing libraries on LTL solvers
June and July	Testing with multiple cases of navigation tasks, refine representation
August	Fabrication of hardware for experiments
September and October	Automated generation of fabrication plans, hardware experiments
November	Compare hardware experiments to design expectations, refine software
December	Final experiments and fine-tuning, report writing

4 Required Resources

1. Computational: A computer capable of running C++ and Arduino IDE software
2. Fabrication: Fabrication would require 3D printing custom parts designed in our system or sourcing compatible existing parts. In addition it will require access to hand tools for assembly.
3. Note that the above resources are available either in Dr. Cynthia Sung's laboratory or in other MEAM department facilities.

References

[1] Calin Belta, Antonio Bicchi, Magnus Egerstedt, Emilio Frazzoli, Eric Klavins, and George Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics and Automation Magazine*, 14(1):61?70, 2007.

[2] S. Castro, S. Koehler, and H. Kress-Gazit. High-level control of modular robots. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[3] Edmund M. Clarke and Jeannette M. Wing. Formal methods: state of the art and future directions. *ACM Computing Surveys*, 28(4):626?643, Jan 1996.

[4] A. Hall. Seven myths of formal methods. *IEEE Software*, 7(5):11?19, 1990.

[5] Gangyuan Jing, Tarik Tosun, Mark Yim, and Hadas Kress-Gazit. An end-to-end system for accomplishing tasks with modular robots. *Robotics: Science and Systems XII*, 2016.