Configuration and Controller Verification for SMORES modular robot

January 12, 2015

Abstract

Key points:

- Build complex configurations and controllers from a library of basic configurations and controllers by arranging them in (nested) parallel and series structures
- Both script and graphical user interface methods are implemented to build configurations and controllers
- Verify that there is no self collision in composed configurations and during the execution of controllers without simulation in Gazebo
- Define "unexpected behaviors" due to instability of the configuration under a controller, and verify that there is no "unexpected behaviors" during the execution of controllers without simulation in Gazebo
- Human asistant configuration and controller design (synthesis).
- Show experiment in simulator. (Maybe with realy robot)

1 Action Items

- Check literatures about quasi stability - Tarik
- Define unexpected behaviors - Jim
- Get final draft at the beginning of Januray - All

- Think about how meta-module is related with this paper - Tarik
- Motion planner for modular robot - Jim
- Come up with set of configurations and controllers - Shangyi

2 Introduction

- Introduce existing designs on modular robots.
- Introduce existing works on modular robot controller design
- Introduce SMORES modular robot and advantages
- Introduce the contribution of this paper
- Why fully automomous approach does not work well so far

3 Preliminary

SMORES robot module Define the ability of motion and connectivity of a SMORES robot module. Position and velocity of each Dof. Can be connected to four other modules at the same time. Representation of properties of a SMORES module, e.g. joint angels, global positions, connection information

Configuration Define the representation of a configuration as a set of SMORES robot modules connected in a certain way. Define topology graph.

Controller Define controller as a basic feedback controller for each Dof of each SMORES module. The reference input of the controller is a gait table. Define how the gait table is executed. Mention that controller in this paper refers to the input gait table.

Collision Define a collision between SMORES modules.

Controller conflict Define a conflict between controllers, i.e. giving opposite commands to the same Dof of a module at the same time.

Unexpected behavior Define the unexpected behavior of a configuration due to instability during a controller execution.

4 Approach and Algorithm

Configuration composition Define the composition of a set of configurations to a single configuration.

4.1 Configuration Composition

Input A set of configurations. A topology graph representing the connectivity among those configurations. A base module (for position transformation).

Output A composed configuration if it is safe.

Procedure

- Start from the configurations that connects to the configuration with base module, transform their positions based on the position of the base configuration and topology graph.
- Check if there is any collisions among the modules and report such collision.
- **Check if the final configuration is stable. If not, find the plane that will make the configuration stable and transform the configuration.**
- Show the expected behavior in simulator.

Controller composition Define the composition of a set of controllers to a single controller. Define the difference between a parallel composition and a series composition. Define the control composition graph.

4.2 Controller Composition

Input A configurations. A set of controllers. A control composition graph.

Output A composed controller if it is safe.

Procedure

- Compose the set of controllers based on the given control composition graph. Explain how the parallel composition and series composition are handled.
- **Check there is no controller conflict in the composition.**
- Execute the composed controller in user defined incremental time interval. At each time step, update each module position and check collision.
- **At each time step, check if the configuration will not have any unexpected behavior.**

4.3 Complexity

Discuss the complexity of the algorithm with respect to the number of modules and size of gait tables.

5 Example and Experiment

With simulation in Gazebo:

- Show a configuration composed from a set of basic configurations.
- Show a composed controller that results in a collision in the configuration.
- Show an updated controller that resolves the collision
- Show a composed controller that results in an unexpected behavior.
- Show an updated controller that eliminates the unexpected behavior.

6 Conclusions

We worked hard, and had fun.

7 Future

- How to represent different attribute/ability of the configurations