

Self-Adaptive Furniture with a Modular Robot

Chih-Han Yu*, François-Xavier Willems, Kristina Haller, and Radhika Nagpal
Self-organizing System Research Group
Harvard University, USA

Donald Ingber
Department of Surgery & Pathology
Harvard Medical School, USA



Figure 1: The self-balancing table. The table legs are formed by modules that are programmed to jointly maintain the table surface level.

Abstract

Robotics has great potential in designing interactive furniture. Modular robots are a special type of robot that are composed of multiple units, and each of them has independent computation and actuation capabilities. Recently, we have proposed an *environmentally-adaptive* shape formation framework for modular robots [Yu et al. 2007]. In this paper, we review one specific application under this framework—a self-balancing table. The table legs are formed by connected modules that are programmed to jointly maintain the table surface level at all time.

We present a decentralized control algorithm to achieve this goal which has several salient features: first, it relies on simple local rules for each module; second, if the environment changes, the modules automatically adapt to jointly maintain the table surface level; third, it is scalable to the number of modules and finally it is robust to individual module failure. In addition, such an algorithm can achieve a wide range of environmentally-adaptive structures beyond the level table surface.

There are several ongoing research directions in our group that share the same theme. We foresee such types of *self-adaptive* frameworks have great potential in future interactive/adaptive furniture designs.

1 Introduction

Interactive domestic design has gained much attention in several traditional computer science fields, including robotics, artificial intelligence, and human-computer interaction. Several successful interactive furniture applications are based on hardware and controller design principles in robotics. For example, D’Andrea et al. incorporate a mobile (wheeled) robot in the design of “The Robotic Chair.” to allow the chair to have self-assembly capability [D’Andrea et al. 2006]. In addition, a vision system is used to guide the chair in establishing its legs.

Modular robots are a class of robotic systems composed of many identical, connected, programmable modules that can coordinate to change the shape of the overall robot [Rus et al. 2002]. In this paper, we present a self-balancing table that is based on a modular robot framework. The modules are connected in a chain to form the legs of the table. They are programmed to cooperatively maintain the table surface level (or a particular orientation) irrespective

of the underlying environment. Since the modules are identical, a module’s function can be easily interchanged with that of another module. This allows modular robots to be more robust to individual component failure than the traditional robotic systems. Along with this merit comes with a challenge: Since a modular robot may be composed of many modules, the control algorithm needs to be scalable to the number of modules.

The controller design of the self-balancing table is based on the decentralized shape formation algorithm previously proposed in [Yu et al. 2007]. Instead of a centralized controller that coordinates and controls all modules, each module autonomously cooperates with its neighboring modules to achieve a goal. The local control algorithm run on each of the modules has the following merits: (1) it relies on simple, local rules for each module, (2) it is highly responsive. If the environment changes, the global configuration autonomously adapts, (3) it is both scalable to the number of modules and robust to individual module failures, (4) the distributed control is provably correct: it provably converges to the desired goal state, i.e. the level table surface. In addition, the upper bound on convergence speed can be theoretically characterized, (5) it is robust towards the sensing and actuation noise in our hardware experiments, (6) the concept of an environmentally-adaptive shape can be generalized to a wider range of domestic applications, such as a chair with a seat back that can autonomously deform to different shapes.

The remainder of this paper is organized as follows: In Section 2, we briefly recap the decentralized control framework proposed in [Yu et al. 2007]. In Sections 3 and 4, we describe the hardware implementation of the self-balancing table and experimental results. In Section 5, we present potential applications of this framework and the ongoing effort towards this direction in our research group.

2 Decentralized Control Framework

In this section, we briefly recap the modular robot model and the decentralized shape formation algorithm. We note that the table surface of the self-balancing table is a special case of environmentally-adaptive shapes in which the shape is programmed to maintain a level surface at all times.

Module Model

Modules can contain four components. They are:

- **Computation:** All modules have identical programs, but may

*Corresponding e-mail: chyuu@fas.harvard.edu

have different roles and thus behave differently. We assume that the computation power of a single module is limited, and our focus is on simple local rules that do not require complex calculations.

- **Communication:** Each module can communicate with its immediate physically-connected neighbors.
- **Actuation:** Each module is equipped with an actuator that allows it to change its angle.
- **Sensing:** Several modules are equipped with accelerometers that allow the module to determine its tilt angle with respect to the environment.

Decentralized Shape Formation Algorithm

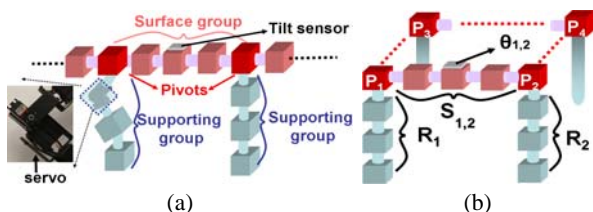


Figure 2: (a) Flexible surface model. Each module is equipped with a servo, and one surface group module is also equipped with a tilt sensor. (b) Group identity initialization in table structure, with pivots (P_i), surface groups ($S_{i,j}$), supporting groups (R_i), and tilt sensors ($\theta_{i,j}$).

Group Initialization: In [Yu et al. 2007], modules are connected to form a flexible surface with supporting legs (e.g. Fig. 2 (a)). Thus the modules belong to one of two classes of groups: the *surface groups* and the *supporting groups*. The flexible surface is formed by all surface groups. A supporting group is a leg of the robot. *Pivots* are a special type of modules which horizontally connect to surface modules and vertically connect to supporting modules (as shown in Fig. 2 (b)).

Surface Specification: Given our flexible surface, we define a shape by specifying the desired tilt angles of all surface groups (inter-pivot tilt angles). In the case of self-balancing table (level table surface), all desired tilt angles are equal to zero (as shown in the following figure).

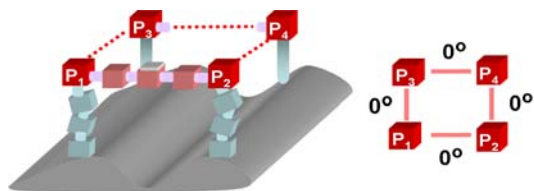


Figure 3: An example of surface specification

Distributed Feedback Control: Here we give a brief overview of the distributed feedback control algorithm: once a desired shape has been specified, all modules coordinate to deform the current shape until the desired goal is reached. As shown in Fig. 4, we can think of the algorithm as continually iterating between two steps.

In Step 1, the sensor on each surface group $S_{i,j}$ transmits its tilt angle $\theta_{i,j}$ to the neighboring pivots P_i and P_j . Each pivot P_i collects tilt sensor information from neighboring surface groups and computes the *aggregated feedback*.

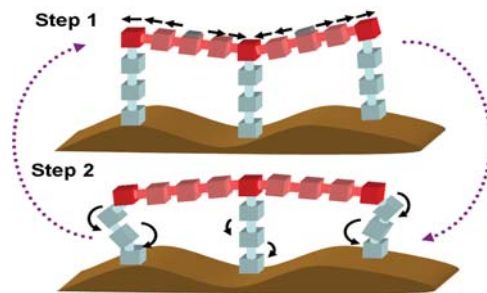


Figure 4: Step 1: Sensor on each surface group propagates current tilt angle to neighboring pivots. Step 2: Based on aggregated feedback computed by its pivot, each module in supporting group responds to the feedback by controlling its actuator.

In Step 2, each supporting group R_i receives an aggregated feedback from its pivot. Each module in the supporting group uses this information to control its actuation. When the desired shape has been achieved (aggregated feedback is close to zero), modules no longer change actuation. If the environment changes, however, modules will respond automatically.

Each module has only a local view, and each of them has its own goal (achieving desired local shapes). When the whole robot is placed on a new rough terrain, each module will not be able to achieve its respective goal via merely one time actuation (since all modules are simultaneously changing their configurations). This iterative sensing-actuation distributed control scheme allows modules to collaboratively achieve their goals. For details of the algorithm please refer to [Yu et al. 2007].

3 Hardware Design

As shown in Fig. 5 (a), the self-balancing table is composed of four supporting groups (legs), and each composed of three modules. Modules in each leg are connected in a chain u-bracket mechanism. Each module is equipped with a Hitec standard servo which can perform a rotation of 90° in either a clockwise or counterclockwise direction. Since the table surface is designed to be flat, the surface group modules described in previous section are replaced by a single rigid surface that forms the table. We mounted a two-axis (x and y) tilt sensor (Analogue Devices ADXL311 accelerometer) on the table surface. Each of the pivots can receive from this sensor, instead of having their own tilt sensors.

For simplicity of implementation, the distributed shape formation algorithm is run on a laptop computer that simulates purely distributed control. Although the distributed control is simulated, our hardware implements the sensing and distributed actuation so that we can directly test our framework in the face of real-world noise. After each module computes the new angle of its servo, the control signal is sent to the hardware robot via serial port. It takes approximately 50 milliseconds for all modules to finish one iteration.

4 Experimental Results

In the first experiment, we examine how quickly and accurately the robot responds to consistent, rapid environmental changes. In this experiment, we fix the robot's four supporting groups to a rigid board. We repeatedly changed the orientation of the board to examine the robot's response. One additional tilt sensor is mounted on the board to record environmental changes. This sensor does not supply input to the robot. Empirically, the sensors we use are somewhat noisy.

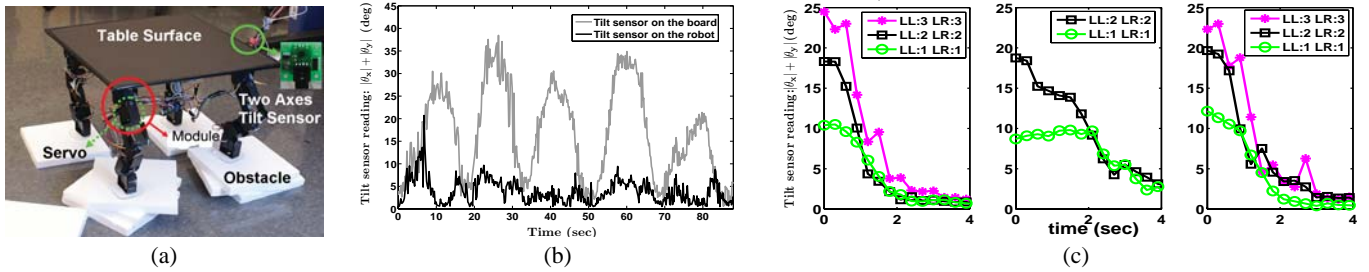


Figure 5: (a) A self-balancing table robot. Each module controls a rotary servo. A two axes tilt sensor (accelerometer) is mounted on the table surface. (b) The robot’s response time to repeated environment changes. (c) Robustness test results when one of the modules does not respond. The left, middle, and right figures are scenarios when top, middle, and bottom modules do not respond respectively.

Fig. 5 (b) shows the results of the experiment. Y axis represents the tilt angle of the surface and x axis represents time. We can see that even when the tilt angle of the floor is changed by $30^\circ - 40^\circ$ over a few seconds, the table is able to quickly respond and keep the surface level. The table never tilts more than $5^\circ - 8^\circ$ after the initial correction.

We also perform robustness experiments by observing the robot’s reaction when one of the modules fails. We tested it under different tasks and in different module failures. We tested two situations which a module fails to respond: (1) the module’s servo is disabled and becomes a passive link, so it freely takes on any angle with no resistance to movement; and (2) the module’s servo remains stuck at the zero degree position at all times. We discovered that the first case does not affect the effectiveness of the algorithm, while the second case affects a few scenarios.

Fig. 5 (c) shows the table’s responding time to achieve levelness while different modules do not participate the task. We lifted one side of the table to 1 to 3 bricks high respectively. At each height, one of the three modules in the supporting groups that needs to be compressed is disabled. We repeated this process four times and Fig. 5 (c) shows the average of robot’s tilt angle across time. We can see from the middle figure of Fig. 5 (c) that the middle module’s failure is more critical than top and bottom modules.

5 Other Interactive Furniture & Conclusion

We are also exploring other interesting applications under this framework. Here we illustrate several examples:

Terrain-Adaptive Bridge: In our framework, one can achieve a modular robotic bridge that can adapt to different terrains. We constructed a terrain-adaptive bridge simulator with Open Dynamics Engine (Fig. 6 (a)). When it is placed on an unknown rough terrain, the robot can automatically form a flat surface or a smooth incline. Even if the terrain changes over time, the modular robot adapts to maintain a level surface.

Smooth Walkway: As shown in Fig. 6 (b), one can potentially build a robotic walkway with this framework. When encountering mountainous terrain, such system can “roll” over the terrain to form a smooth incline.

User Adaptive Chair: In addition to environmentally-adaptive structure, our framework can potentially achieve interactive/dynamic shape formation. One example application is a chair with its seat back capable of deforming, creating various shapes to fit the needs of different users or to heal the users’ back disorders (as shown in Fig. 6 (c)).

3D Relief display: 3D physical display is an application where a

modular robot forms arbitrary shapes as a novel form of 3D media and visualization. Our proposed flexible surface can act as a “relief” display, since the distributed algorithm can easily achieve complex shapes (as shown in Fig. 6 (d)). Applications in this domain require efficient shape transformation. The distributed property of our algorithm makes this high dimensional control problem scalable and provides such efficiency.

In addition to the above illustrated examples, this adaptive structure framework can be extended to achieve more functional structures through designing different feedback and shape formation mechanisms. We believe such *self-adaptive* framework has the potential in many interactive domestic applications. Further information please visit: <http://www.eecs.harvard.edu/~chyu/ModularRobot>

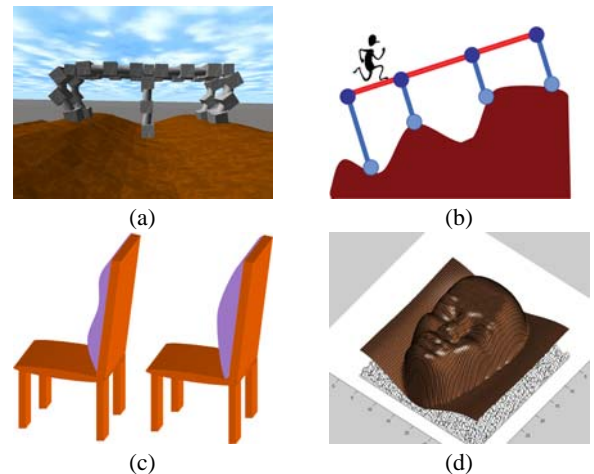


Figure 6: (a) Terrain-adaptive bridge (b) Smooth walkway (c) User adaptive chair (d) 3D relief display.

References

D’ANDREA, R., DEAN, M., AND DONOVAN, M. 2006. The robotic chair. In *IdeaCity06*.

RUS, D., BUTLER, Z., KOTAY, K., AND VONA, M. 2002. Self-reconfiguring robots. *Communications of the ACM*, 2002.

YU, C.-H., WILLEMS, F.-X., INGBER, D., AND NAGPAL, R. 2007. Self-organization of environmentally-adaptive shapes on a modular robot. In *Proc. of IROS*, 2007.