Submitted to the 1997 IEEE International Symposium on Computational intelligence in Robotics and Automation (CIRA '97)

Extending the Teleo-Reactive Paradigm for Robotic Agent Task Control Using Zadehan (Fuzzy) Logic

Edward P. Katz

Visiting Scholar
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, CA 94305
katz@flamingo.stanford.edu

and

Product Generation Information Systems
Hewlett-Packard Company
1501 Page Mill Road, MS 4U6
Palo Alto, California 94304
ekatz@corp.hp.com

***Abstract**: The Teleo-Reactive (T-R) paradigm for computing and organizing actions for autonomous agents in dynamic environments can be viewed as having a foundation triplet of durative actions, circuit semantics, and boolean logic. This foundation is intended to enable the construction of semantic circuitry for the continuous computation of parameters and conditions on which agent action is based. Drawing on the notion of analog circuitry, actions and behaviors are intended to be more continuous over periods of time. However, the use of boolean logic can sometimes inadvertently revert behaviors to be reminiscent of the state transition model since its limited bivalent capability is not consistent with the continuous nature of durative actions and circuit semantics. This document is a report of work in progress on a Teleo-Reactive extension where Zadehan (fuzzy) logic is substituted for boolean logic.*

1. Introduction

The Teleo-Reactive (T-R) paradigm for agent control [Nilsson94, Nilsson92] is influenced by the need for autonomous agents, such as mobile robots to "operate in dynamic and uncertain environments". Nilsson recognizes that the standard methods of artificial intelligence based on explicit declarative representation and reasoning processes are strongly challenged by these environments. He also cites alternative approaches that directly relate sensory inputs to actions and shares with control-theory the notion that continuous environmental feedback is necessary for effective action. The intent of T-R is to focus on importing some control-theory ideas into computer science.

An agent control program is a Teleo-Reactive sequence that directs the agent toward a goal (hence *teleo*) while at the same time responding (hence *reactive*) to the perceived dynamic environmental changes in circumstances that the agent encounters. Drawing on the notion of analog circuitry, T-R is intended to provide an agent task execution environment having a more continuous nature than one based on a state transition model. Consistent with these notions is the continuous-valued logic of Zadeh [Zadeh65, Mendel95]. This paper presents work in progress on an extension to the conventional T-R paradigm using Zadehan (fuzzy) logic. The extension is intended to enhance the T-R capabilities beyond boolean logic where it is currently limited.

The paper organization will first present an overview of the Teleo-Reactive paradigm highlighting the essential concepts and issues. Next are discussions for the motivation and description of this proposed extension. Finally, a brief description of some early simulation results will be described.

2. Background

The Teleo-Reactive (T-R) paradigm is conceived to be a kind of *middleware* for agent programming (whether robotic or software agent). The *upperware* would contain the high level planning activities needed to determine what the agent should do and to create the corresponding T-R sequence(s). Likewise, the *lowerware* would process the T-R action commands in order to realize the effects of these action commands (e.g. decoding *move* and *rotate* actions into motor voltages to turn the wheels). T-R is strictly intended as a middleground in between. Should an unanticipated situation arise for which the available T-R sequence(s) can not handle, the upper level planning activities would be invoked to generate a new T-R sequence capable of negotiating the new situation.

Among the several concepts embodied in the T-R paradigm are the T-R sequence and the semantic circuitry. A T-R sequence resembles a set of ordered production rules in its simplest form (figure 1a). The conditions (K_i's) are boolean expressions containing predicates about sensory inputs and other information from the world model and are continuously computed with K_1 considered the goal condition. The highest true condition K_i (i.e. smallest index i) determines the rule to be "fired" (executed) and is known as the first true condition (FTC). The actions, a_i's, which have some effect on the environment or world model are *durative* rather than discrete in that the action continues indefinitely. If a T-R sequence contains just one rule K_1 **->** a_1, then a_1 would be energized as long as K_1 is true and ceases when K_1 becomes false. If a sequence contains more than one rule, the action of the rule corresponding to the FTC is energized and continues until that rule is no longer the FTC. This may occur if either the FTC becomes false or if the condition of another higher (smaller index) rule becomes the FTC. These can be induced by changes that the actions make to the environment or model, unanticipated effects of the actions, or unexpected exogenous world events. The actions, a_i's, themselves, can either be primitive actions, a set of concurrent actions or they can be T-R sequences themselves providing hierarchical capability. It is important to realize that *all* conditions at *all* levels of the hierarchy are continuously being evaluated.

A simple T-R sequence (figure 1a) can be thought to be *compiled* into and executed from semantic circuitry (figure 1b). The circuity is driven by condition-computing circuitry that is continually computing predicate values from the environment consisting of sensors and world model. In both figures, all conditions, K_i, are continuously computed. As Nilsson [Nilsson94] describes,

> *The condition K_1 is taken to be the goal condition, and the corresponding action, a_1, is the null action. The condition K_2 is the weakest condition such that when it is satisfied (and K_1 is not), the durative execution of a_2 will (all other things being equal) eventually achieve K_1. And so on. Each non-null action, a_i, is supposed to achieve a condition, K_j, strictly higher in the list ($j<i$).*

From the diagram in figure 1b, the negation of the condition K_1 "signal" inhibits any lower actions from being energized. Thus, if K_1 is true, no lower rule action (higher index) can be active. If K_1 is not true, then likewise, the highest condition K_j which is true will "inhibit" all the lower rules, K_{j+1}, ..., K_n activation. Thus in a semantic circuit, a condition, K_j, energizes its corresponding action, a_j, if it is true and there are no higher level conditions (K_1, ..., K_{j-1}) that are also true. K_j also energizes an "inhibit" signal to all lower level rules preventing those actions (a_{j+1}, ..., a_n) from being energized. The inhibit signal of the highest true condition is the semantic circuit's mechanism for implementing the FTC by disabling the lower rules.

2.1 An Example

A simple example of robots moving bars within a two-dimensional world [Nilsson94] can illustrate these concepts (figure 2). Suppose in a two-dimensional simulated world, robots can move, rotate clockwise, grab bars, and move bars to various locations. Now suppose a robot is to move to the point *bar center* in order grab bar A. Suppose, also, that the robot can sense its environment and can evaluate conditions which tell it whether or not it is already grabbing bar A (*is-grabbing*), facing toward bar A (*facing-bar*), positioned with respect to bar A so that it can reach and grab the bar (*at-bar-center*), on the perpendicular bisector of bar A (*on-bar-midline*), and facing a zone on the perpendicular bisector of bar A from which it would be appropriate to move toward bar A (*facing-midline-zone*). Also, suppose the robot is capable of executing primitive actions *grab-bar, move*, and *rotate* (clockwise) with the obvious effects. Execution of the T-R sequence in figure 2a will result in the robot moving to and grabbing bar A if it is not already.

Notice how each properly executed action in this sequence achieves the condition in the rule above it, eventually leading toward the goal. This example uses only primitive actions. It should be re-emphasized that each action may be a primitive action, another T-R sequence, or a set of concurrent actions (including a mixture of primitive and T-R sequence). Hierarchical and recursive programs are allowed building T-R trees. The interested reader may pursue these topics in Nilsson's work [Nilsson94].

Suppose when the T-R sequence (figure 2a) is executed, the robot and environment is as portrayed (figure 2b). For example, suppose initially only the last (or default) rule is true. The last rule becomes the designated FTC and causes its primitive action, *rotate,* to start the robot a clockwise rotation about its axis and continues until the robot is **exactly** facing the midline-zone. Recognition of this zone causes the predicate *facing-midline-zone* to become true as well as becoming the FTC. The new FTC's corresponding action, *move*, is now energized while the *rotate* action ceases. The robot continues forward until it **precisely** reaches the midline-zone causing the condition *on-bar-midline* to become true. The current *move* action is disabled and the corresponding *rotate* action begins. Processing continues in this way for the other rules until finally the robot is holding the bar.

For sake of illustration, suppose in the course of this execution, an unforeseen exogenous environmental "event" occurs in which a strong wind "blows" the robot a great distance and leaves it in a new orientation. Because many of the rule predicates have changed, a new FTC is immediately found, its corresponding action energized, and execution continues in a like manner from the new position and orientation. The point to be made here is that what appears to be state transition behavior is actually not the case. Rather, the transitioning from one rule to another is based strictly upon changing predicate values.

3. Motivation

Binary predicates can sometimes be a constraint when dealing with real world sensor readings and perceived conditions which may not exhibit crisp outputs. This imposes one or more interpretation thresholds which may require adjustment to achieve the desired behavior. Often ranges of sensor reading have to be identified as belonging to certain predicates, and even then, the thresholds between the different ranges can sometimes be merely arbitrary and abrupt.

The first true condition (FTC) constraint is useful in the crisp Teleo-Reactive paradigm, especially where the programs are T-R trees and branches represent alternative (disjunctive) subgoals. But it also restricts any activities to the currently energized action or action set for the FTC rule. Sometimes it is desirable to overlap the current actions with the previous actions or subsequent actions for smooth transitions between actions.

In the robot and bars example above, there are many assumptions made for that scenario. Among them are that all sensing and positions are precise, movements are accurate, etc. This usually won't be the case for real world applications.

4. Extension Using Zadehan (Fuzzy) Logic

The Teleo-Reactive paradigm can be viewed as having a foundation triplet of durative actions, circuit semantics, and boolean logic. Intuitively, these are strange bed fellows. Durative actions and circuit semantics are intended to provide a notion of continuity. It seems that the "odd" element (boolean logic), while usable, causes T-R sequences to behave somewhat reminiscent of a finite-state device. For example, a particular rule containing the current FTC is active until another higher condition becomes the FTC. This in turn causes the new rule to "fire", energizing its action and inhibiting any lower rule actions from being energized. This behavior might appear similar to that of one based upon a state transition model and not particularly consistent with the Teleo-Reactive paradigm intent.

Substituting boolean logic and its binary predicates with fuzzy logic and its continuously-valued counterparts provides a more consistent foundation triplet. Since fuzzy logic is a generalization of boolean logic, all desirable aspects of crisp T-R are preserved.

4.1 Modification 1: Continuous-valued Predicates

Suppose the constraint of binary predicates is relaxed and extended so that predicates now have a range over the continuous interval [0,1], where in addition to being FALSE or TRUE (represented by 0 and 1 respectively), they can also be partially true by exhibiting a degree of truth (truth strength). For example, this could allow a position predicate (e.g. *at-bar-midline*) to also indicate the degree of nearness if its strength is positive but less than 1. In effect μ(*at-bar-midline*) is a shorthand notation for the truth value of the fuzzy predicate μ(*position* is *at-bar-midline*).

Referring to figure 1b, let the negation (~) and conjunction (∧) functions in the semantic circuit be replaced by the corresponding Zadehan negation and T-norm. This time in the robot and bars example (figure 2) the predicate values are assigned the truth values of the fuzzy set membership functions. In applying these extensions to our example, the robot's forward sensor may have some sensitivity distribution which could function as a kind of peripheral vision. It might have precise sensitivity directly in front of the robot and drop off significantly off to the side. When the T-R sequence (figure 2a) begins again as depicted (figure 2b), the robot is initially rotating clockwise on its axis. This time as the robot begins to face the midline zone, the forward sensor begins to "partially" sense it. When this happens, the *facing-midline-zone* predicate now has a strength value μ(*facing-midline-zone*), 0<μ(*facing-midline-zone*)≤1, which represents the degree to which the forward sensor registers the midline-zone directly in front of the robot.

4.2 Modification 2: Functional Actions

If predicates can have continuous valued truth strengths, then how should actions function when a predicate is only true to a degree? Suppose actions are allowed to have the *potential* for acting in a manner that is related to the corresponding condition's strength $\mu(K_i)$. That is, suppose actions are functional in the sense that they have an argument in addition to any other arguments which has the truth strength value of the corresponding condition (i.e. $a_i(\mu(K_i))$).

In the crisp version of example (figure 2a), the *move* and *rotate* actions are assumed to have fixed rates of linear and rotational velocity respectively. Suppose now they are functional actions using the corresponding truth strength to indicate a proportional effect. For example, if a_i is a *move* action, then it *might* become a straight-line move at a speed linearly proportion to $\mu(K_i)$ (e.g. linear-velocity=$\mu(K_i)$ x maximum-linear-velocity). In effect, this is an example of how an action *could* be interpreted as "partially" energized to a degree, though this particular interpretation is not a requirement.

While this might have desirable intuitive appeal, one must also consider the consequences. It would seem reasonable that *sometimes* there should be a direct relation between the a condition's truth strength, $\mu(K_i)$, and the corresponding degree that action, a_i, should be applied. Other times, the predicate should be binary. Emulation of binary predicates via fuzzy set singletons can be used where needed as in the case of the *is-grabbing* binary predicate (since partially holding a bar has little meaning).

4.3 Modification 3. Eliminating the First True Condition Constraint

Suppose in the example that μ(*facing-midline-zone*) = 0.25 and all other higher conditions are false (i.e. $\mu(K_i) = 0$, $1 \le i < 6$). If both continuously-valued predicates and their corresponding "partially" energized actions are allowed, it can be seen from the circuit semantics diagram (figure 1b) that the bottom rule will only be partially inhibited (i.e. ~ μ(*facing-midline-zone*) = 1 - μ(*facing-midline-zone*) = 0.75). This contradicts the FTC constraint as there can now be two or more rules whose actions are partially energized. Eliminating this constraint allows for several (or potentially all) rules to be active, each to a different degree (which is also the case for fuzzy expert rule-based systems). Continuing with our example, and allowing both *move* and *rotate* to be functional having proportional respective velocities to their corresponding $\mu(K_i)$'s, the rotating robot, which is now beginning to perceive the midline-zone (μ(*facing-midline-zone*)>0), begins to perform two things at once. First, with an increasing inhibit signal (~ (μ(*facing-midline-zone*)>0)) generated from the above rule, the bottom rule's rotation rate begins to slow. The more the robot faces the midline zone, the slower the rotation. Second, even though the robot is not directly facing the midline-zone, it begins to slowly move forward. This process continues with the rotational velocity decreasing and the linear velocity increasing. As the robot nears the *midline-zone*, the *on-bar-midline* predicate begins to have support (μ(*on-bar-midline*)>0) and it's corresponding *rotate* action partially energizes. The actual performance of the different actions can be modified by adjusting the membership function shapes for the different predicates.

The observant reader has probably recognized that eventually several *move* actions and several *rotate* actions can each be energized to some degree concurrently. This is resolved by applying separate defuzzification algorithms. One defuzzifier returns a crisp move velocity, and the other a crisp rotate velocity. The environment is updated and new predicate values are computed and the processing continues. By removing the FTC constraint, multiple actions of the same nature (e.g. several *move* actions) can be potentially be "blended"[Saffiotti93] via defuzzification.

5. Fuzzy-Teleo-Reactive

The Fuzzy-Teleo-Reactive (F-T-R) extension embodying these three modifications is a natural extension to the Teleo-Reactive paradigm. In the case of the robot and bars example of figure 2 where all the actions are primitive, it can be observed that this simplest form is somewhat similar to a fuzzy Sugeno-Takagi (S-T) type system [Babuska94] where the fuzzy rules are of the form

$$\texttt{if}\ x_1\ \textit{is}\ A^i_1 \text{ and } x_2\ \textit{is}\ A^i_2 \text{ and } \ldots\ x_n\ \textit{is}\ A^i_n\ \texttt{then}\ y=F^i(.)$$

and the *i*th rule's conclusion, $F^i(.)$, is some linear function. The dot in the function argument denotes that the arguments can be variables (inputs) in the rule premises (x_i's) as well as others (e.g. environment variables), y is the output, and A^i 's are fuzzy sets defined by membership functions.

The F-T-R sequence of figure 2a can be expressed in a way that **`resembles`** a Sugeno-Takagi type system:

`if` *is-grabbing* `then` ***nil***

`if` ~ *is-grabbing* and (*at-bar-center* and *facing-bar)* `then` ***grab-bar()***

`if` ~ *is-grabbing* and ~ (*at-bar-center* and *facing-bar)* and (*on-bar-midline* and *facing-bar*) `then` ***move()***

`if` ~ *is-grabbing* and ~ (*at-bar-center* and *facing-bar)* and ~ (*on-bar-midline* and *facing-bar*)
and *on-bar-midline* `then` ***rotate()***

`if` ~ *is-grabbing* and ~ (*at-bar-center* and *facing-bar)* and ~ (*on-bar-midline* and *facing-bar*)
and ~ *on-bar-midline* and *facing-midline-zone* `then` ***move()***

`if` true `then` ***rotate()***

But an S-T type system is not sufficient to support F-T-R in general. In an T-R (F-T-R) sequence, each action can be primitive, a T-R (F-T-R) sequence, or a set of concurrent actions (each of which may be primitive or a T-R (F-T-R) sequence). Thus, a non-primitive action can not be represented by some linear function $F^i(.)$.

6. Current Status

This paper is a work in progress report of the current Fuzzy-Teleo-Reactive extension. More investigation is needed in several aspects of this extension. Currently, a series of simulators are being built beginning with Botworld [Teo95] the two dimensional simulated robot world. The most recent simulator uses Fuzzy-CLIPS [Orchard95] to assist with the fuzzy processing. Successful simulation results will lead to trials using actual mobile robots. Figures 3a and 3b illustrate the results of some early simulations. In the first case, the membership function for *facing-midline-zone* is wider than in the second case. This results in the first case in the robot starting to move forward earlier as its rotating and “overshooting” the midline zone. In the second case, the membership functions for *facing-midline-zone* is narrower and *on-bar-line id wider.* This causes the robot to pass near, but not through, the midline zone on its way to the bar center.

The F-T-R extension is evolving to incorporate the capability for non-primitive actions and is the subject of the next report. The early simulation trials have shown that the predicate’s membership function shapes can significantly affect this ability and without care can potentially produce undesirable performance. Other future areas of investigation include learning F-T-R rules and shapes of the membership functions. The membership functions could possibly be temporal membership functions whose shapes change over time (possibly in response to changes in the environment or conditions of the robot).

7. CONCLUSIONS

This paper is a work in progress report which describes an example of enhancing the Teleo-Reactive paradigm by drawing upon concepts found in fuzzy logic. This particular approach for a Fuzzy-Teleo-Reactive paradigm described in this paper suggests that boolean predicates be replaced with continuous-valued ones, that the effects of actions *may* be functional with respect to the corresponding condition strength, and that the relaxation of the first true condition (FTC) constraint can in this paradigm enhance capabilities. The combination of these three modifications allows for a more parallel evaluation capability and potentially more robust functionality while still preserving the fundamental continuous intent of the original paradigm.

Acknowledgments

This investigation effort has benefited greatly from the collaboration and encouragement of Nils Nilsson. Alexander Shoykhet, who performed early simulation trials using Botworld [Teo95] in his senior year programming project, was the first to demonstrate some of the F-T-R’s basic principles.

References

1.[Babuska94] Babuska, R., Jager, R., and Verbruggen, H.B., *Interpolation issues in Sugeno-Takagi Reasoning*, **IEEE International Conference on Fuzzy Systems**, 1994, pp. 859-863.

2.[Nilsson92] Nilsson, Nils J., *Toward Agent Programs with Circuit Semantics*, Tech. Rep. STAN-CS-92-1412, **Department of Computer Science**, Stanford University, 1992.

3.[Nilsson94] Nilsson, Nils J., *Teleo-Reactive Programs for Agent Control*, **Journal of Artificial Intelligence Research**, Vol. 1, 1994, pp. 134-158.

4.[Mendel95] Mendel, J.M., *Fuzzy Logic systems for Engineering: A Tutorial*, **Proceedings of the IEEE**, V. 83, No. 3, pp. 345-77, March 1995.

5.[Orchard95] Orchard, R., *FuzzyCLIPS Version 6.04 User's Guide*, **National Research Council of Canada** Report ERB-1045, June 1995.

6.[Saffiotti93] Saffiotti, A., Ruspini, E., and Konolige, K. *A Fuzzy Controller for Flakey, an Autonomous Mobile Robot*, Technical Note 529, **AI Center**, **SRI International**, 333 Ravenswood Ave., Menlo Park, CA 94025, April 1993.

7.[Teo95] Teo, Patrick Cheng-San and Benson, Scott, *BOTWORLD 3.2*, (unpublished), Robotics Laboratory, **Department of Computer Science**, Stanford University, 1995.

8.[Zadeh65] Zadeh, L.A., *Fuzzy Sets*, **Information and Control**, Vol. 8, pp. 338-353, 1965.

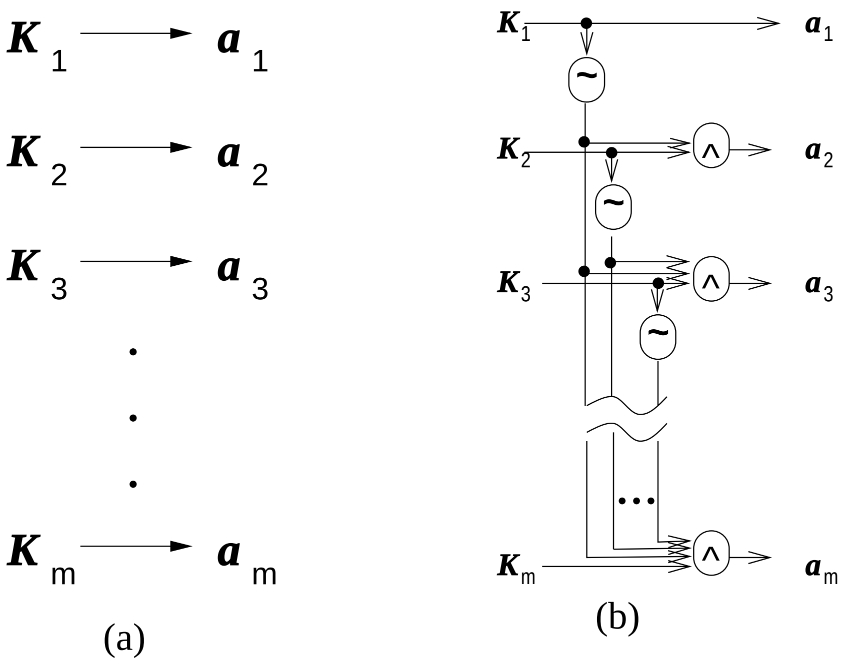

FIGURE 1. (a) A General T-R sequence. (b) Corresponding Semantic Circuit

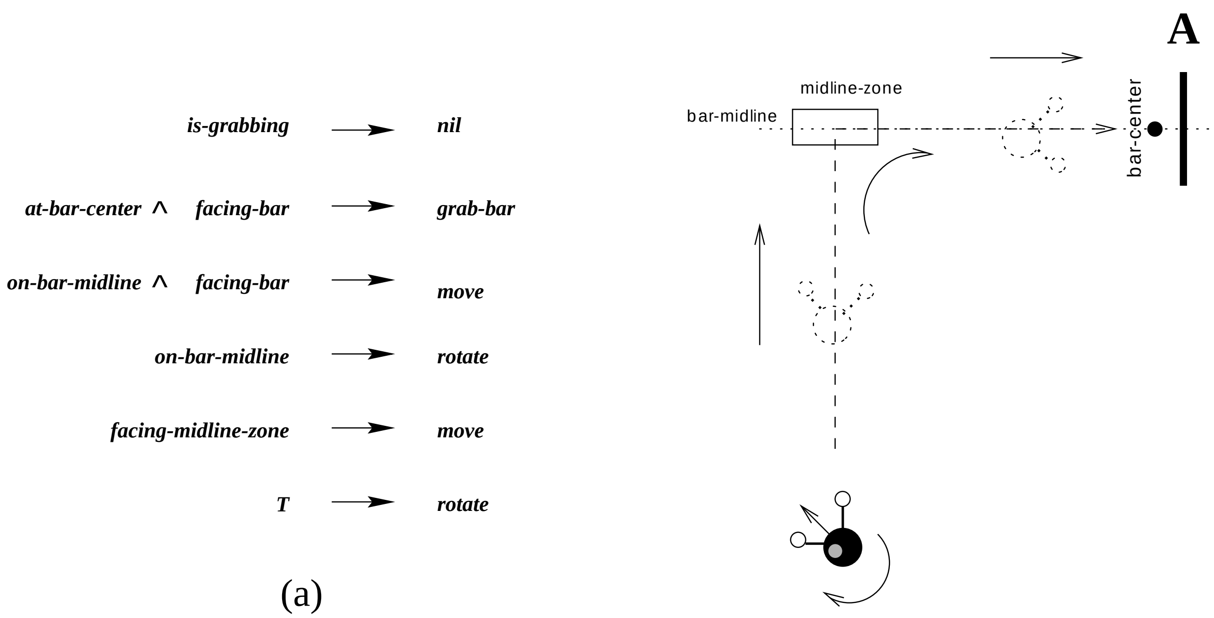

FIGURE 2. Robots and Bars. (a) T-R Sequence (b) Botworld Configuration

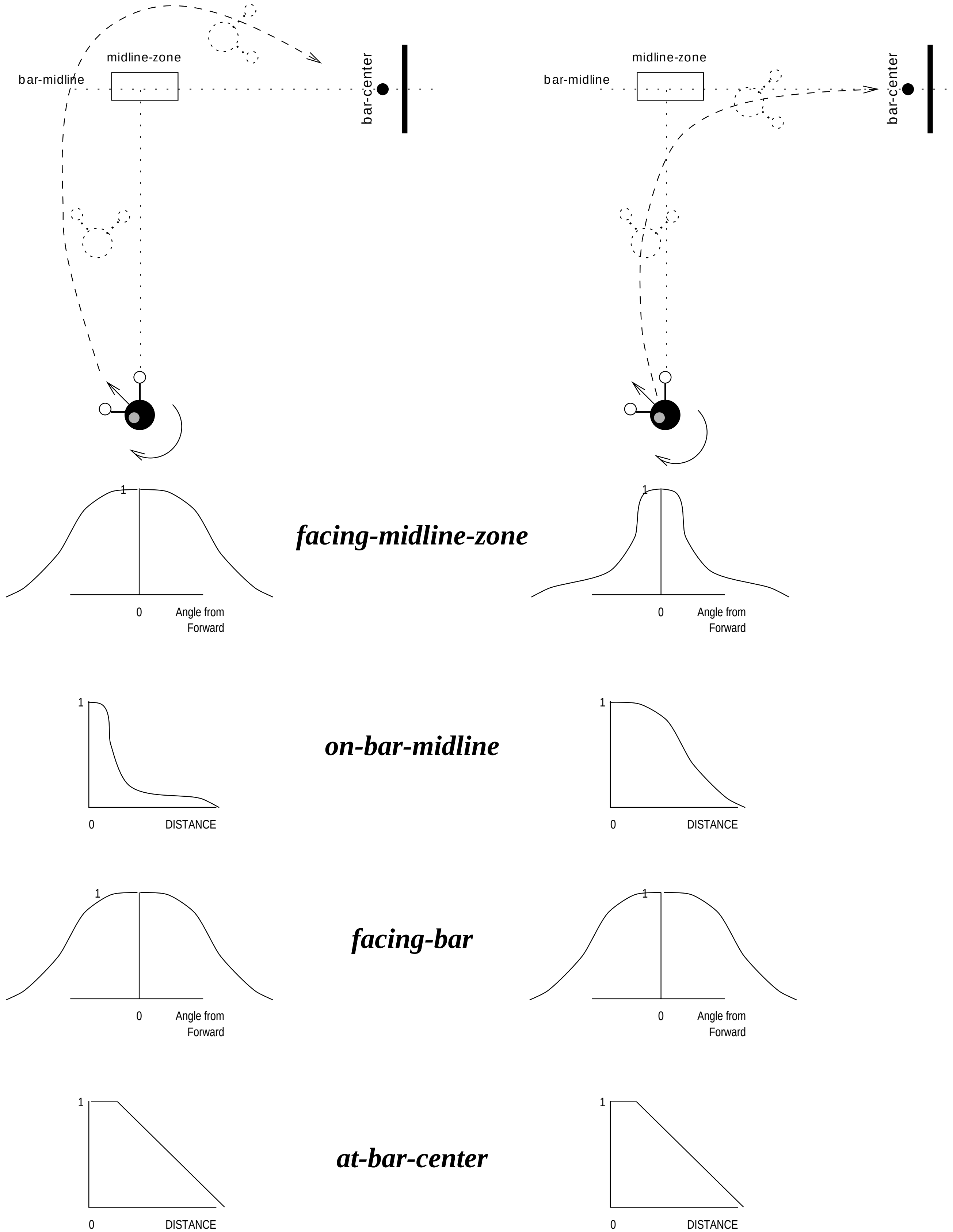

FIGURE 3. Sample experiment runs.