Monitoring Strategies for Adaptive Periodic Control in Behavior-based Robotic Systems

Ernesto Burattini, Alberto Finzi, Silvia Rossi
Dipartimento di Scienze Fisiche
University of Naples “Federico II”
Naples, Italy
{ernb,finzi,srossi}@na.infn.it

Maria Carla Staffa
Dipartimento di Informatica e Sistemistica
University of Naples “Federico II”
Naples, Italy
mariacarla.staffa@unina.it

Abstract—The main goal of our current research is the design of a behavior-based robotic architecture that has the capability of adapting its behaviors to the rate of change of both the environment and its internal states reducing the computational costs of input processing. Inspired by research on biological clocks, we introduced a simple schema theory model where releasing mechanisms are combined with adaptive internal clocks. In this paper, we design the description and development of a complete robotic architecture implementing this model. In particular, we considered a mobile robot domain that simulates the navigation behavior of a Cataglyphis ant enhanced with simple visual capabilities.

I. INTRODUCTION

The main goal of our current research is the design of a behavior-based robotic architecture that has the capability of adapting its behaviors both to the rate of change of the environment and to changes of its internal states. This adaptive behavior requires effective mechanisms for controlling sensors and effectors with respect to the internal constraints imposed by the control system and the external environment. The executive control is to combine different low-level strategies (such as obstacles avoidance, walls follow, gates crossing, etc.), with high-level activities (such as achieving a goal, returning to the nest, etc.), giving them, from time to time, different priority values both for allocation of resources and for action selection processes. The low-level activities affect the safety of the system, and may be achieved by applying principles of reactive control. However, high-level activities generally are achieved by processing more complex tasks, and, therefore, require high computational costs for both the inputs processing and the acquisition of data from the environment. In this context, self-regulating mechanisms balancing sensors elaboration and behavior execution play a crucial role. Indeed, frequent sensors readings could result in a high computational charge that degrades the performances of executive and deliberative processes, while, adopting large latencies in sensors sampling is unsafe since the environment may change too much in between two consecutive readings. In this sense, any effort to build a cognitive architecture for dealing with dynamical and flexible behaviors has to deal with an efficient processing capability of the sensors elaboration.

In a previous paper, Burattini and Rossi proposed the concept of periodical/timed activations of behaviors [1], [2]. Inspired by internal self-regulation mechanisms in living organisms [3], they introduced internal clocks to tune and adapt the frequencies of the behaviors activities. Such mechanisms can speed up or slow down the period of behaviors activation and thereby the reading frequency of the sensors according to both the robot–environment interactions and the interactions that may arise within the robots itself [4], [5], [6].

In this paper, inspired from this work, we propose a real robot implementation of a behavior-based control architecture endowed with periodic releasing mechanisms. In particular, we investigate the feasibility and effectiveness of the approach and we explore how the adaptive clocks affect the computational load for the elaborations of perceptual data, considering each single behavior and the overall system. The question we have to answer is twofold. First, is the robot we are building reliable and safe? This means that the robot should be able to timely react to the external stimuli avoiding dangerous situations. Second, does the introduction of adaptive clocks in a single behavior provide better performances of the robotic system, in terms of a reduction of computational resources spent to achieve the tasks?

To test the proposed model we implemented a behavior-based control system endowed with these regulation mechanisms and evaluated different settings. These mechanisms should provide an interconnecting layer between adaptive behaviors and high level cognition [7].

II. TIMED ACTIVATION OF BEHAVIORS: RHYTHMIC RELEASES

Our aim is to develop a behavior-based control for a robotic system capable of adapting its emergent behavior to the surrounding environment and to its internal state. Our working hypothesis is that adaptive behaviors can be simulated in the control activity of a robot starting from self-regulating mechanisms. In particular, our architecture combines innate releasing or inhibiting mechanism and biological clocks.

a) Innate releasing or inhibiting mechanisms (IRMs): Lorentz [8] and Tinbergen [9] identified in many animals an innate releasing or inhibiting mechanism (IRM) able to control and coordinate behaviors. An IRM presupposes a specific stimulus that releases a pattern of actions. For example, a prey animal may have, as an IRM, the stimulus coming from
The view of the predator, which activates the escape behavior. IRMs were included in the schema representation of behaviors [10] in the form of releasers, controlling when behaviors must be activated or deactivated. A releaser is an activation mechanism that depends on exogenous factors (e.g., presence of a predator) and/or endogenous factors (e.g., hunger).

b) Adaptive Innate Releasing Mechanisms (AIRMs): The releaser’s function, somehow, recalls the notion of “internal clock”, already introduced in some approaches [11], [5], [6] in order to activate motivational states for a robot (for example, hunger or sleep). In fact, an internal clock, similarly to a releaser, represents an internal mechanism which regulates behaviors activations depending on endogenous and/or exogenous factors. However, there are substantial differences between IRMs and internal clocks. Indeed, while a releaser is an instantaneous activation mechanism, the internal clock is periodical and adaptive.

In [1], [2], the authors inspired by the internal clocks, connected the concept of IRM to the concept of a periodical activation of behaviors introducing the Adaptive Innate Releasing Mechanisms (AIRMs). An AIRM is a releasing mechanism endowed with and adaptive period of the releaser activation.

IRMs and AIRMs are illustrated and compared in Figure 1 deploying a schema theory representation [10]. Here, for each behavior, we have a schema composed of a Perceptual Schema (PS), Motor Schema (MS), and an activation mechanism, either a clock or a releasing function \( \rho(t) \). On the left side, IRM enables/disables data flow \( \sigma(t) \) from PS to MS, the MS is activated only in the presence of the stimulus, but sensing data are always processed (i.e., at each machine cycle); in contrast, on the right side, AIRM directly enables/disables data flow \( \sigma(t) \) from sensors to PS, therefore when the activation is disabled sensing data are not processed (sensors readings minimization). Furthermore, the internal clock regulates the frequency of the activations, hence the frequency of data processing (behavior adaptation), using a feedback mechanism on the processed sensing data \( \sigma(t) \). The AIRM mechanism will be detailed in the next sections.

III. THE ARCHITECTURE

The robotic system is controlled by a behavior-based executive where each behavior can be described by a schema theory model [10]. Each behavior of the Robotic System (RS) is provided with a periodical and adaptive clock that controls its activations (AIRMs as defined in [1], [2]). In particular, we assume that (see Figure 1):

- Each clock has a base period (or initial rhythm) called \( p_b \), ranging in the interval \([p_{bmin}, p_{bmax}]\). The values of \( p_b \) and range are experimentally tuned. \( p_b \) depends on several factors such as the sensors precision, environmental conditions and purposes of the behavior;
- The releasing function \( \rho(t) \), is a delta function that, with period \( p_b(t) \), is equal to 1 at each \( p_b \) machine clocks and 0 otherwise, and tells the robot when the PS has to process inputs. This timed releasing function takes data from a PS and returns an enabling/disabling signal to the Perceptual Schema itself;
- \( \sigma(t) \) is the actual function representing the data perceived by the sensor at each time step \( t \);
- \( \sigma(t) \) is the function that represents data coming from sensors at each sampling step: \( \sigma(t) = f(\sigma(t))x\rho(t) \), i.e. \( f(\sigma(t)) \) is for processed sensing data and \( \rho \) is the releasing function.
- \( \pi(t) \) is the command sent by the MS to actuators. While the PS is inactive, no new commands will be sent. This means that an actuator can keep its output constant until a new command will come (e.g., the velocity of the wheels of a robot), or its output will produce an action with its own duration and then it waits for a new command (e.g., a gripper that lifts a box).

Each time the behavior is activated, thus enabling sensory data flow, sensory data are passed in feedback to the internal clock which updates its period depending on internal and environmental conditions. The mechanism for period/rate regulation is called the updating policy and will be detailed in the following sections.

IV. MONITORING STRATEGIES AND ADAPTATION

The emergent behavior will result from a combination of the following parameters:
- the initial period \( p_0 \);
- the range of allowed values for the period \([p_{bmin}, p_{bmax}]\);
- the updating policy.

In particular, the combination of these parameters defines a monitoring strategy. A monitoring strategy is a policy for scheduling sensing activities and has to balance the cost of monitoring and the risk of inaccurate and partial information about the environment.

We will compare our monitoring strategy (adaptive and periodical) with respect to other relevant monitoring strategies proposed in literature. In particular, we consider four different
cases: a) the robot is not equipped with any internal clock; b) the robot is equipped with internal clocks and has a priori knowledge about the task to achieve (for example about the distance to cover); c) the robot is equipped with clocks, but does not have any a priori knowledge about the environment; d) the robot is equipped with internal clocks but not adaptive.

To illustrate these cases we introduce the following example. Let us consider a robotic system whose purpose is to cover a certain distance (goTo behavior).

$$\text{dist} \propto \log_2(\text{dist}) + \frac{(\text{dist}_{\text{cov}})}{p_b},$$

where $$\text{dist}_{\text{cov}}$$ is the distance already covered by the robot.

We can see that the number of activation in the case (b) will have as an upper bound the case (a), and as a lower bound the case (c).

Overall, the benefits brought by adaptive and periodical monitoring strategies are mainly two:

- **periodical mechanisms** of activation can reduce the number of activations of the behavior (with respect to the standard case (a) in which the activations are performed at every machine cycle), causing a relative decrease in the computational burden, and improving performance of the entire system;
- the use of **adaptive mechanisms** allows us to obtain a behavior that adapts itself to the specific environmental conditions (e.g. the robot reads sensors more often if there is a dangerous situation and less often in cases of a safe operational situation).

In the following, we will show that, by calibrating appropriately the basic rhythms and using the appropriate policies to update them, we can obtain a significant improvement in performance compared to an architecture without rhythms.

**V. IMPLEMENTATION**

Based on the model introduced in the previous sections, we designed a system whose behavior is mainly guided by the visual information in a 3D environment, constituted by an office space, that implements the three monitoring strategies described here. We used a PIONEER 3DX provided with a blob camera and sonars. The robot is controlled by a Player/Stage client [14].

c) **Cataglyphis domain:** We evaluated our approach using a mobile robot that simulate the navigation behavior of a Cataglyphis ant enhanced with simple visual capabilities. The robot has the task of searching food in its environment without any a priori knowledge and then return to its nest following a straight path, without taking into account the trajectory followed during the searching of food. The Cataglyphis domain is a good testbed for our purposes: the domain is interesting from a behavioral point of view and well analyzed in several ethological field trials [15]; the ant is provided with internal mechanisms such as guidance and dead-reckoning; we can associate with this testbed all the monitoring strategies previously presented (constants, with a priori knowledge, without a priori knowledge).

d) **Behavior-based control:** The robot behavior is obtained as the combination of the following primitive behaviors (see Figure 3): AVOID, WANDER, PATH_INTEGRATION, MOVE_TO_FOOD, MOVE_TO_NEST and FIND_LANDMARKS.
organized in two meta-behaviors called LOOK\_FOR\_FOOD and RETURN\_TO\_NEST.

![Control architecture for the mobile robot in the Cataglyphis domain.](image)

**e) System assessment:** To assess the system performances, we compared the adaptive control system with respect to a not adaptive one (i.e. with sensor readings fixed at every machine cycle). Our aim is to show that a significant improvement in performance can be obtained by appropriately tuning the basic periods of clocks and the policies to update them. Therefore, for each specific behavior we will evaluate the updating policies, seeking, on one hand, to optimize performances (i.e. less sensor readings), and, on the other hand, to enhance the robot safety and the correctness of the overall system behavior.

**f) Behaviors settings: For each behavior, we have to define the monitoring strategy composed of the 1) updating policy and the 2) base period, that is empirically defined after a phase of testing.**

1) **Updating policies:** We start describing each behavior and the associated updating policy.

The WANDER behavior provides a random search in the environment. Its output consists only in a random direction for the robot. Since the activation of this behavior is periodic, but not adaptive, WANDER can be associated with a constant clock. That is, the updating policy is the following: \( p_\beta(t) = p_b = \text{const}, \) where, the clock period \( p_\beta \) remains equal to the base period \( p_b. \)

In contrast, the AVOID behavior, responsible for obstacle avoidance and its output consists in a direction and sets the navigation velocity for the robot. This behavior is safety critical and needs an adaptive clock and an associated updating policy to timely react to dangerous situation. In this case, a good policy is to change the AVOID clock period according to the first derivative of the sensory input \( \sigma(t) \), that represents the distance from the nearest object, evaluated by the sonars. More formally, the clock period \( p_\beta \) is updated as follows:

\[
p_\beta(t) = \varphi(\frac{\sigma(t)-\sigma(t-p_0)}{p_0} \times \rho(t)),
\]

where \( \rho(t) \) is the period at the previous behavior activation (initially \( p_\beta(0) = p_0 \)), \( \rho(t) \) is the releasing function that enables the sensory sampling, \( \varphi \) is a normalizing function mapping the derivative into a set of permitted values and \( \sigma(t - p_0) \) is the value of \( \sigma \) at the previous behavior activation. Intuitively, the clock frequency is adaptive with respect to the environmental changes: the higher the change the smaller the sensory sampling.

**MOVE\_TO\_FOOD** detects the food and guides the robotic system towards it, setting its direction. In order to obtain reliable information from the camera, this behavior forces the robot to slow down its velocity. By choosing an adaptive clock period for this behavior (i.e. reducing the number of activations), we allow the robot to reach the food as soon as possible. However, we have to set the base period and the updating policy taking into account that we want to reach the goal with the minimum docking error. Thus, we have to balance effectiveness and precision. The idea is to keep constant the base period \( p_\beta(t) \) until the robot reaches a fixed distance threshold and then to decrease the period linearly with the food distance as follows:

\[
p_\beta(t) = \varphi(\text{distance}(t) \times \rho(t)),
\]

where \( \varphi \) is a normalizing function, \( \text{distance}(t) \) is the distance at time \( t \), \( \rho(t) \) is the releasing function that enables sensory sampling. Here, the behavior performances will depend on the optimal balance between the clock frequency and distance.

**PATH\_INTEGRATION** manages the direction and the distance to return to nest. This behavior uses odometric sensor information to calculate the robot shift w.r.t. the previous position. Analogously to WANDER the behavior remains always active with a constant clock period, until it reaches the food. In the return process, the behavior is activated only after an abrupt change of course, e.g. due to the presence of obstacles, which diverts the trajectory of the robot that has to be recalculated. Therefore, we can state that \( p_\beta(t) = p_b \times \rho(t). \)

**RETURN\_TO\_NEST** set the direction of the robot toward the nest. Like MOVE\_TO\_FOOD it requires reliable information from the camera and so will slow down the robot velocity while its PS is active. This behavior exploits an a priori knowledge (i.e. the distance to the nest) that allows us to set the base period value proportional to half of the distance from the nest, \( p_b = \text{distanceToNest}(t_0/(2k)) \) and then reduced accordingly, i.e. \( p_\beta(t) = \varphi(\text{distanceToNest}(t)/\rho(t)) \). It has been shown that, asymptotically, this strategy is very efficient [12], [13].

2) **Setting the base period:** Once we have defined the behaviors’ schemata and the associated updating policies, it remains to set the base periods \( p_b. \) These are empirically defined after a phase of testing. In the following, we detail the case of MOVE\_TO\_FOOD.

As previously mentioned, the updating policy is to keep constant the clock period until a certain distance threshold is reached, then it should be linearly reduced. During the exploration of the environment, the robot can detect the food at different distances, hence we need different distance threshold. In order to optimize the performances and simultaneously
Finally, we noticed that if the robot detects, for the first time, from the target, hence not incurring in dangerous situations, there are settings where the robot can stop at a safety distance. A horizontal line separates safe/unsafe settings: above the line, the number of activations and the total amount of time spent in behavior simulation is 0.283 ± 2.83e−004, because it allows to minimize the time of behavior activation with a very small docking error. Moreover, for each case, a horizontal line separates safe/unsafe settings: above the line, the robot can stop at a safety distance from the target, hence not incurring in dangerous situations. Finally, we noticed that if the robot detects, for the first time, the food at a distance smaller than 60cm, the period has to reach the minimum value immediately. Looking at the docking error and the total amount of time for the behavior activations obtained in each test, we can choose the best period/distance that balances the tradeoff between safety and efficiency.

VI. EMPIRICAL RESULTS

In order to assess the system performances, we compared the system behavior when endowed with adaptive clocks with respect to a not adaptive periodic version (e.g., activations at the machine clock). In particular, for each behavior, we considered the number of activations and the total amount of time spent in behavior execution. In Figure 5, we show some results from the analysis of the \textsc{return_to_nest} behavior, where for each case we plot the results of one trial. We note that the number activations of this behavior, with an adaptive clock, is proportional to $\log_2(\text{dist})$ whereas, in the case of non adaptive clocks, these increase linearly with time. Furthermore, we note a significant reduction of the execution time of the behavior endowed with the adaptive clock.

If we consider the overall system we observe a considerable advantage in performances in the case of adaptive clocks. This is illustrated in Figure 6 where we compare the performances in term of activations for different behaviors (i.e., with different adaptation strategies) considering the overall system. We notice that the adaptive monitoring strategy of \textsc{return_to_nest} (i.e., with a priori knowledge) produces better results in terms of number of activations.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, we explored the feasibility and the effectiveness of a behavior-based robotic architecture where the behavior activities are modulated by periodic and adaptive releasers. In particular, inspired by internal self-regulation mechanisms in living organisms, we introduced internal clocks to tune and adapt the frequencies of the sensors and activations associated with the system behaviors. Such mechanisms can speed up or slow down the period of behaviors activation and thereby the reading frequency of the sensors according to both the robot-environment interactions and the interaction that may arise within the robots itself. To test the model we implemented a behavior-based control system endowed with these mechanisms and evaluated different monitoring strategies. In particular, we developed an application which simulates the behavior of the Cataglyphs ant.
Behavior based robotic usually resolves conflicts by deploying a subsumption architecture or by implementing some control mechanisms in order to switch between tasks, selecting the action [16], [17]. For example, in [18] the authors presented a schema theoretic model for a praying mantis which behaviors are driven by motivational variables such as fear, hunger and sex-drive. In this approach, the action selection module selects only the motivational variable with the highest value. In our approach, the modulation of behavior is not controlled by an on/off switch for changing the task. Indeed, the global behavior emerges from the each single behavior through a rhythmic controller modulated by a monitoring strategy.

Other authors dealt with this kind of problems. For example, in [11] the authors presented a parallel architecture focused on the concept of activity level of each schema which determines the priority of its thread of execution. A more active perceptual schema can process the visual input more quickly and a more active motor schema can send more commands to the motor controller. However, while in our approach such effects are obtained through rhythmic activation of behaviors, in [11] the variables are elaborated through a fuzzy based command fusion mechanism.

In this research, we performed a preliminary assessment of an architecture in which the activities of each behavior are triggered by a clock. In our experiments, we observed that the number of activations of each behavior decreases strongly, i.e. the computational overload is lowered. Furthermore, our tests have also shown interesting results about the synchronization and the scheduling of the behaviors. In fact, since each behavior is endowed with its own clock, whose period changes are based on external and internal conditions, we have that, in some circumstances, a behavior activated more frequently than others with the consequence that its influence on the emergent behavior is stronger than other behaviors. We observed that our architecture is only partially reactive since we store at each time the values of the clocks for each behavior and sensor data. In fact, the mechanism of adaptive clock provides both a quick reaction to the perceived stimuli and takes into account the rate of changing in the external environment and internal states. In other words, our system refers, in some way, to the history of the rhythms fluctuations caused by the changes of the environment and of the internal state. This means that in a pure reactive system we have the same action in presence of the same stimuli while in our system the action depends also on previous stimuli.

In future works, we plan to investigate the introduction of learning mechanism to select the proper rhythms for each behavior. Moreover, in this work we assumed a setting where each behavior modulates its own rhythm of activation, however the behaviors interconnection can be enhanced, in this direction we plan to investigate how the adaptive periodical activations of behaviors may influence and constrain each other.

![Number of behaviors activations with or without adaptive clocks.](image)

**ACKNOWLEDGMENT**

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no.216239.

**REFERENCES**


