**Corresponding Author:**
Matthew J. O'Brien, Mobile Robot Lab, Georgia Institute of Technology,
Technology Square Research Building, Room S27, 85 Fifth St. NW,
Atlanta, GA, 30308, USA
Email: mjobrien@gatech.edu

# Adapting to Environmental Dynamics with an Artificial Circadian System

Matthew J. O'Brien[1]. Ronald C. Arkin[2]

[1]Department of ECE, Georgia Institute of Technology, Atlanta, GA

[2]College of Computing, Georgia Institute of Technology, Atlanta, GA

## Abstract

One of the core challenges of long-term autonomy is the environmental dynamics that agents must interact with. Some of these dynamics are driven by reliable cyclic processes, and thus are predictable. The most dominant of these is the daily solar cycle which drives both natural phenomena like weather, as well as the activity of animals and humans. Circadian clocks are a wide-spread solution in nature to help organisms adapt to these dynamics, and demonstrate that many organisms benefit from maintaining simple models of their environments and how they change. Drawing inspiration from circadian systems, this work models relevant environmental states as times series, allowing for forecasts of the state to be generated without any knowledge of the underlying physics. These forecasts are treated as special percepts in a behavior-based architecture; providing estimates of the future state rather than measurements of the current state. They are incorporated into an ethologically-based action-selection mechanism, where they influence the activation levels of behaviors. The approach was validated on a simulated agricultural task: a solar-powered agent monitoring pest populations. By using the artificial circadian system to leverage the forecasted state, the agent was able to improve performance and energy management.

## Keywords

long-term autonomy, behavior-based robotics, circadian rhythms, action-selection

## Introduction

Moving from structured, static environments to natural, dynamic environments remains a challenge in robotics. Reactive approaches in behavior-based designs have shown effectiveness in dealing with the randomness present in the real world, but most ignore the somewhat predictable dynamics that many environments exhibit. Drawing inspiration from circadian rhythms in nature, this work describes an approach for modeling environmental dynamics and adapting an agent's behavior to them.

The need for such an architecture in robotic systems is growing. More than ever, robotic agents are leaving controlled factory floors and entering complex dynamic worlds. Recently, the non-industrial robotic market overtook the industrial robotic market for the first time ("The Robotics Industry Will Reach $237 Billion in Revenue Worldwide by 2022", 2017). These platforms are entering homes, fields, and city streets. They must deal with varying weather, light, and activity from both humans and animals. The widespread use of circadian clocks in natural organisms, from plants to humans to even bacteria (Paranjpe & Sharma, 2005), suggests that robotic agents would also benefit from some circadian-like system to adapt to their environment's dynamics.

This work is particularly targeted for slow and persistent robotic agents. The benefits of persistent robots are clear, and slow robots may have several advantages, especially for persistent energy-constrained tasks (Arkin & Egerstedt, 2016). We use the terms "slow" and "persistent" in relation to the speed of change in the environment. The most common examples will be daily cycles, where significant changes generally take on the order of hours. A persistent agent persists through significant change in its environment. As opposed to an agent that might execute for an hour and see few differences. Similarly, a slow agent acts on roughly the same time scale that the environment sees meaningful change. For example, if some relevant event happens over an hour, an agent that can always respond within five minutes can easily react to it. An agent that needs 45 minutes or more will struggle. If the slow agent can anticipate the event, or other changes in its environment, it can take action ahead of time and bypass some of its limitations.

In this paper, we develop an artificial circadian system (ACS) to learn and exploit the patterns that often exist in the dynamics of natural environments. In previous work, we demonstrated the application of time series modeling of an environment for a robotic agent and tested it on a simulated robot interacting with pedestrian traffic using simple behavioral rules (O'Brien & Arkin, 2017). This research extends that work with a principled method to incorporate the generated predictions into action-selection.

The follow sections in this paper will first overview circadian systems in nature and related work in robotics and artificial intelligence. Then the artificial circadian system will be presented. Next a simulated experiment providing an example application on a notional agriculture task will be detailed. The results of the experiment, and a discussion of those results, follows. Finally, the paper concludes with a summary of the work, discussion around when the artificial circadian system best applies, and proposed future work.

## Related Work

### Circadian Rhythms

In many organisms, there exists a special set of biological rhythms. These "circa-rhythms" each corresponds to a unique temporal niche associated with some geo-physical process (Aschoff, 1967). The most well known and studied being circadian rhythms, which correspond to the daily solar cycle. Circadian rhythms are the original inspiration for this work, and their widespread use in nature within organisms at all levels of complexity suggests that some circadian-like system would also benefit robotic agents.

A circadian clock has three primary components. First is an internal oscillator which tracks the passage of time and defines the approximate period of an environmental cycle. This internal oscillation is created by a Transcription Translation Oscillating Loop (TTOL). A feedback loop between gene expression and protein production, whose details are out of scope of this work (see Richard & Gumz, 2013). Second is a sensory input channel which allows the oscillator to entrain to the environment through light, impacting the TTOL. Finally, there is the output channel, where the current state of the circadian oscillator drives both metabolic processes and behavior.

Circadian clocks have been observed to offer different types of advantages. Some involve the synchronization of metabolic processes within an organism, or behavior between social organisms (Paranjpe & Sharma, 2005). In this way, circadian clocks are leveraged the same as a normal clock. Circadian clocks are even used for navigation in birds (Oatley, 1974). The usage of the circadian clocks relevant to this work is as a method for synchronizing to a changing environment. Some related species of parasites are active at different times of day: their circadian clock is used to avoid competition (Fleury, Allemand, Vavre, Fouillet, & Bouletreau, 2000). There are also examples where timing activity helps avoid predation (DeCoursey, 1997). Rather than simply reacting to environmental change, circadian clocks allow an organism to take action before its senses (perceptual state) or needs (internal state) could drive the behavior. “Circadian rhythmicity of behavior represents an animal’s information, or one is tempted to say, knowledge about a particular feature of its environment… and what to do about it.” (Oatley, 1974 p. 456)

This work is biologically-inspired, but does not attempt biomimicry. The mechanisms of a circadian clock are not copied. Instead, two main principles are extracted. First that simple, data-driven models are very useful in complex, noisy environments. Circadian systems demonstrate that an agent or organism can gain value from having even simple models of complex processes in the environment. If the underlying physics of a system can be modeled directly, e.g. as a dynamical system, that could be a superior approach. However, doing so for complex natural environments may be impossible, or at least impractical, on an autonomous robot. Second, that entrainment is a key property of circadian systems. In nature, a “free-running” circadian clock often has a notable difference in period from the environmental cycle it is tracking. Similar to a mechanical oscillator, entrainment drives the circadian oscillator at the relevant environmental frequency and ensures the circadian system will resync even after extreme disturbances (Roenneberg, Daan, & Merrow, 2003). If the environmental dynamics were completely deterministic and reliable, entrainment would not be necessary. For cyclic change (a focus in this work) the problem may even reduce to finding a static optimal schedule of activity over the cycle. Natural environments are rarely so reliable. Even the presence of light, driven by constant steady planetary rotation, experiences seasonal shifts as well as random perturbations from weather. The approach outlined in this paper is based on the idea of using simple, data-driven models that can be constantly entrained to the dynamics a situated agent observes, as an efficient means to understand of how the environment is changing.

## Robotics and AI

The challenge of a changing environment is fundamental to the goal of long-term autonomy in robotics. Mapping of dynamic environments over time has received considerable attention within long-term autonomy. Most approaches in the literature look to merely filter out changes from the static map, though some approaches do explicitly model the environment's dynamics (e.g. Mitsou & Tzafestas, 2007, Ambrus, Ekekrantz, Folkesson, & Jensfelt, 2016). More relevant is work on modeling environment dynamics that is applied to robot behavior. The most similar research comes from work on the "frequency map enhancement", or FreMEn, system (Krajnik, Fentanes, Santos, & Duckett, 2017). In this approach environmental states are treated as binary variables, whose values are modeled as probabilistic functions over time, represented using Fourier series. These simple and efficient models can be applied to small sections of an environment, allowing for the representation of spatio-temporal dynamics. This was leveraged for several types of problems, including localization and navigation. The FreMEn approach shares many ideas with this research. The main difference being that the methods in this paper allow for continuous state and more than purely cyclic dynamics. This broadens the potential applications, but the increased complexity limits the ability to use them at high resolution to capture differences over space.

Research on energy harvesting for wireless sensor networks has also investigated ways to adapt to changing environments. For sensor nodes, the goal is to reach "energy neutral operation" so that the node can continue to operate for long durations (on the order of years). To achieve this, both predictive forecasts of future energy and reactive responses to the actual gathered energy can be used to adapt the behavior of the sensor node (Kansal, Hsu, Zahedi, & Srivastava, 2007; Buchli, Sutton, Beutel, & Thiele, 2014). In this case, action is limited to adjusting the duty cycle the sensor runs at, changing the amount of monitoring performed and energy expended. There is overlap between the work presented in this paper, and some of the modeling approaches in the sensor network domain. Though the action-selection methods are highly specialized for the limitations of the immobile nodes.

Lastly the domain of reinforcement learning holds interesting potential. The problem of reinforcement learning is to both learn and solve a Markov Decision Problem (MDP): defined as a set of states $S = \{s\}$, actions $A = \{a\}$, transition probabilities between states $T(s, a, s')$, and rewards for acting and reaching a specific state $R(s, a, s')$. The goal of reinforcement learning is to output the optimal policy $\pi(s)$: a mapping from state to action that maximizes the expected reward. This conventional approach has no way to represent a changing environment, but some methods for MDPs have been developed for to incorporate change over time. Time-Dependent MDPs (TMDP) provide a formulation that includes time as a special continuous variable in the state space, and models the durations of actions as dependent on time (Boyan & Littman, 2001). Very recent work instead makes the transition function $T$ time-varying, dubbed Time-Varying MDPs (TVMDPs), which allows forecasted dynamics to be included into it (Liu & Sukhatme, 2018). This work more directly captures the differences between time and other states, and how the environment (modeled as the transition function) changes over time.

# The Approach

The artificial circadian system is built on a behavior-based architecture. A special set of perceptual schemas (or perceptual algorithms that process sensor data for specific behaviors; Arkin, 1998) model the environmental dynamics as time series, and forecast future values of the state. Behaviors are supplemented with activation functions based on both the current measured state and the future predicted state. Action-selection is done by selecting the behavior with the highest activation. Figure 1 shows the top-level architecture, and the rest of this section details the components.

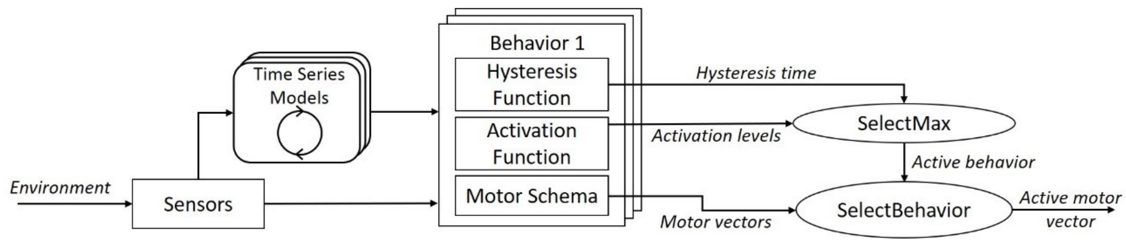


Figure 1. The top-level diagram of the Artificial Circadian System.

## Modeling environmental dynamics as time series

Without an effective approach for understanding how the environment will change, a robotic agent cannot adapt its behavior to said dynamics. We view the environment state as a time series, and apply methods from the time series literature to model and forecast future values. Time series models predict, or "forecast", future values of a variable from its past values. It allows for modeling of both cyclic and non-cyclic effects, and can forecast both the value of the future state as well as generate prediction intervals. A useful quality is that the approach is data driven. The underlying cause of the dynamics may be too complicated to model directly, or at least impractical to do so on a robot (for example, consider the actions of many individuals to create traffic). Time series modeling offers versatility and simplicity, side-stepping the need for expert knowledge about a domain. One of the most direct approaches is the auto-regressive (AR) model shown below in equation 1, where the next value of our variable $y_t$ is the weighted sum of the past values, along with a constant term $c$ and an error term $\varepsilon_t$:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{p-1} + \varepsilon_t \quad (1)$$

A fundamental technique in modeling time series is the classical decomposition (Hyndman, Koehler, Ord, & Snyder, 2008), of a time series into its major components:

$$Y_t = T_t + S_t + E_t \quad (2)$$

Where $Y_t$ is the original time series. $T_t$ is the trend component, a slowly changing average level; $S_t$ is the seasonal component, a repeating pattern with known period; and $E_t$ is the residual, or error, left over after the trend and seasonal components have been removed from the time series. This decomposition provides a structured approach for modeling and allows for focus on components of interest. For example, seasonal effects are sometimes removed from data in a process called deseasonalization. In our work the seasonal effects are of key importance, as they represent the cyclic (or circadian, if the period is roughly 24 hours) dynamics in the environment.

The literature on time series analysis and forecasting provides a broad set of tools (Hyndman, 2017). At this time, however, there is no catch-all method or system. Historical data is required to manually select an appropriate model, one that can capture the dynamics of the environment. While model selection must be done offline, model fitting (i.e. parameter estimation) and forecasting can be done online, autonomously by the agent.

Time series modeling has traditionally been applied for offline analysis of discrete data. To leverage these techniques on a real-time system, time is broken down into discrete periods which we will call forecast intervals. The past values of the state in each forecast interval are stored as the vector $S_P$. This vector is fed into the time series model, and used to forecast the future values of the state, $S_F^k$, where $k$ is the timestep this forecast vector was generated. Lastly, there is a set of measurements for the current forecast interval called $S_M$. At the end of any forecast interval, $S_M$ is used to generate a final state value, $S_P(k)$, for that interval. $S_P(k)$ may be an average of all measurements in $S_M$, a count of recorded events, or even a null value if no measurements of the relevant state variable were made. Figure 2 visualizes how these vectors represent the state over time.

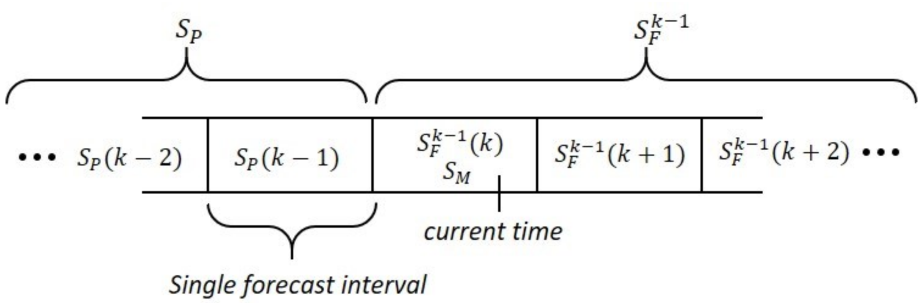


Figure 2. $S_P$ contains measurements of the state in the past, while $S_F$ contains forecasts of the state in the future, including the forecast of the current state made in the past. $S_M$ is the current measured state.

A notion of entrainment to the environment is captured in the system at two levels. Every forecast interval the agent appends a new value to $S_P$, and generates a new forecast vector $S_F^k$. This means the forecast for some specific point in time in the future is repeatedly updated (usually becoming more accurate) as new measurements of the environment are taken. Secondarily the new history, $S_P$, can be used to re-estimate the model parameters. Adapting the time series model if the dynamics of the environment have changed.

## Action-selection via activation levels

The agent has some understanding of how the environment is changing, represented as a vector of forecasts into the future. How should the agent act based on these predictions? The action-selection method used in this work is part of a family of approaches that assigns an activation level to each top-level behavior an agent can execute. Arbitration via action-selection is straightforward: the behavior with the highest activation level is executed. This leaves the question of how to calculate that activation level. Many approaches for this problem have been developed (Richter, Sandamirskaya, & Schöner, 2012; Blumberg, 1994; etc.). The approach in this paper is based on an ethologically-inspired architecture, described in (Arkin, Fujita, Takagi, & Hasegawa, 2003).

Every activation function is broken into three components. The first is a motivation function (MOT) which represents a behavior's internal motivation to activate. This is based on endogenous (internal) variables representing the state of the agent. Such as its power level. The second component is the releasing mechanism (RM). It differs from the motivation function in that it focuses on external, or exogeneous, variables. The RM is a special function that specifies both the conditions required for a behavior to activate, and how well they are satisfied. Algorithmically, if its output is zero it means necessary conditions to execute a behavior are not currently satisfied, and the behavior cannot execute regardless of how high its total activation may be.

In this research, we introduce a new component, the circadian function (CIR). This component represents the influence of an associated forecast on a behavior's activation level. Like the releasing mechanism, the circadian rhythm function focuses on exogenous variables, but considers the impact of their future predicted values, rather than their current measured value. The output of each component is defined to be from zero to one. The full activation function is the weighted sum of these three components. The activation function produces a final activation level used to select which behavior to execute. The activation function of behavior N is:

$$act_N = \begin{cases} 0 & if\ RM\ conditions\ not\ satisfied \\ G * \left(W_M * MOT_N + (1 - W_M)(W_C * CIR_N + (1 - W_C) * RM_N)\right) & else \end{cases} \quad (3)$$

Three parameters are used to conveniently adjust robot behavior, based on the approach taken in (Chernova & Arkin, 2007). The weight $W_M$ ($0 \leq W_M \leq 1$) adjusts the relative influence of the exogenous factors (releasing mechanism and circadian function) and endogenous factors (motivation function). This allows for the robot to become more reactive to its environment, or more focused on its internal needs and goals. A second weight is introduced, $W_C$ ($0 \leq W_C \leq 1$). This weight balances the impact of the circadian function against the releasing mechanism, or more generally sets how much the robot focuses on the current measured state compared to the predicted future state. Finally, a behavior gain, $G$, adjusts the overall activation magnitude of a behavior. $G$ effectively represents the priority between behaviors, determining which will execute given that similar activation from each behavior. Figure 3 below displays the structure of the activation function.

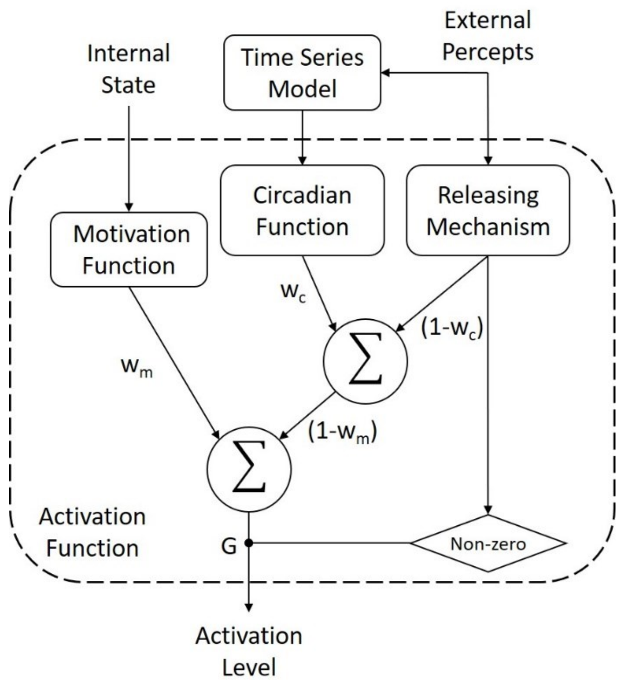


Figure 3. The activation function of a behavior is a weighted sum of the three components, plus a conditional check on the releasing mechanism.

Acting based on any model introduces some level of risk for agents in real environments. A basic level of robustness is achieved in this approach through the redundancy of the releasing mechanism and the circadian function. Both act on the same relevant exogenous variables, with the weight $W_C$ determining the impact from each. If $W_C = 1$, the behavior fully leverages the forecast. This may be the best action if the forecast is accurate, and detrimental if it is not. For all other values of $W_C$, the releasing mechanism ensures the behavior has some activation due to the current measured external state, regardless of what the time series model predicts.

A final consideration is the issue of behavior dithering. When activation levels change smoothly a rapid switching of behaviors can occur, degrading overall performance. Consider an agent that continuously leaves and returns to a charger, as it collects or expends just enough energy for a charge behavior's activation to cross the threshold of "highest activation". Some method of prioritizing the active behavior, to create behavioral persistence, is needed. The most common approaches are inhibition of inactive behaviors by the active behavior (Blumberg, 1990), or hysteresis in the action-selection process (Velayudhan & Arkin, 2017). This approach uses the latter, and each activation function is supplemented with a hysteresis function that defines a minimum amount of time the behavior will execute.

The ACS allows an agent to adapt to the future using models of the world, a trait normally associated with deliberative systems. Yet no deliberation or planning is performed; the approach is fundamentally reactive. The agent "reacts" to the predicted future state. Adapting to the future can involve more than responding directly to the forecast at some point in the future. For instance, choosing an ideal time to execute a behavior over a window of time can be done by comparing the current state against the forecasted state throughout that window. An otherwise unmeasured state can be estimated using forecasts of the current state from measurements in the past. In the following experiment, both of these ideas are leveraged.

# Experimental Validation

To validate this architecture, we set up a simulated experiment to test how incorporating the circadian function impacts an agent's behavior and performance. The scenario notionally represents a small, slow agriculture robot; one that persists within the agricultural environment as a beneficial component of the ecosystem, monitoring and tending to the plants.

The agent's purpose is to monitor the population of a pest over a growing season and identify when the population reaches a critical threshold requiring intervention by a farmer. It must do so while spending a minimal amount of time and energy, hypothetically allowing the agent to work on other tasks, and reducing wear on physical components. The dynamics modeled by the time series will be the pest population and solar irradiance. This will inform when the agent must spend time and energy on monitoring the pest population, and when it should charge.

The general procedure for the experiment is to vary the weight of the circadian function (Wc) for one behavior at a time, and observe how it impacts the agent's performance. The two key behaviors are monitor and charge, which utilize the forecasts for the pest population and solar irradiance respectively. The rest of this section covers how the environment dynamics are generated and modeled, how the robot behaviors were constructed, the full experimental procedure, and other miscellaneous details of the implementation.

## Environment Dynamics and Modeling

The pest population dynamics follow a model based on aphids. Many aphid species exhibit exponential population growth after an initial infestation due to their ability to perform rapid asexual reproduction during some periods of their life cycle. This growth is simulated by inserting pests according to a simple aphid population model (Kindlmann, Arditi, & Dixon, 2004), plus a random noise factor. Pest are distributed randomly over the work space. Figure 4 shows several simulated trajectories of the aphid population.

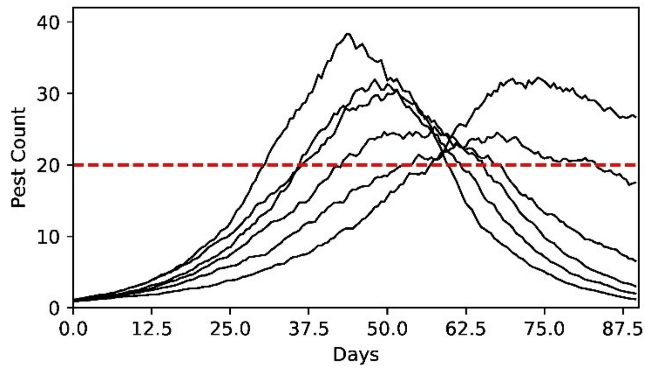


Figure 4. Example population trajectories. The critical threshold for pest population, where intervention is needed, is shown as a dashed red line.

The pest dynamics were stored on the agent as a time series of measured pest levels, defined as the number of pests found per area searched. A 12-hour time interval was used. As the agent often did not monitor in a 12-hour period, values between measured intervals were interpolated, while values after the last measurement were forecasted. The pest dynamics were modeled using the forecast.stlm method in the R forecasting library (Hyndman, 2017). This approach first breaks the series into a trend, seasonal, and stationary components using the STL decomposition (Seasonal and Trend decomposition using Loess: Cleveland, Cleveland, McRae, & Terpenning, 1990). The stationary component was modeled as an ARMA(2,1) process. Forecasts of the overall series are generated by taking the forecast of the stationary component, and reseasonalizing it with the trend and seasonal components. For details on these forecasting techniques, see Hyndman et al. (2008). Nine previous seasons, each 90 days long, were simulated to provide data to fit the model parameters.

Solar irradiance available to the robot was based on measurements from the National Solar Radiation Database (National Renewable Energy Laboratory, 2016). Change in available power is driven primarily by the daily solar cycle, secondarily by weather effects, and finally a small impact due to the seasonal shift in daylight hours over the 90 days the data was drawn from. Figure 5 visualizes the variability in the amount of solar energy available per day.

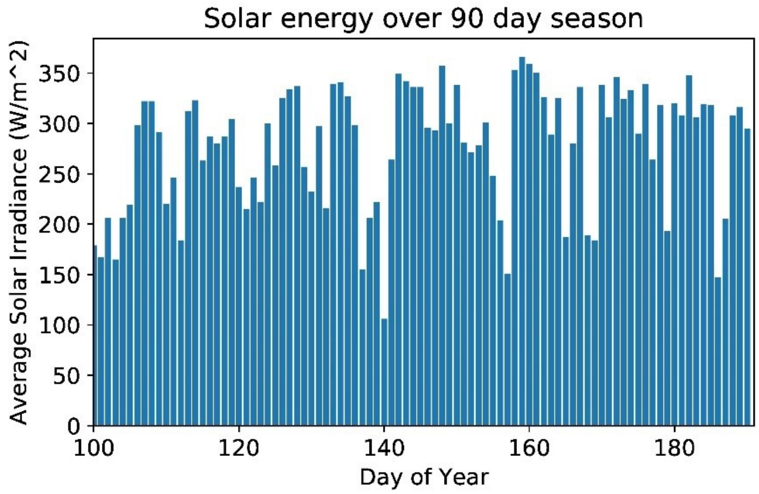


Figure 5. Average daily solar irradiance over 90 days.

Some of the most advanced and widely available forecasts are for weather. While manual forecasting of solar irradiance using time series is possible (e.g. Bacher, Madsen, & Nielsen, 2009), in this experiment we simulate the agent receiving weather forecasts from an outside entity. Forecasts are generated by taking the ground truth measurements and multiplying them with a random factor. This factor increases exponentially the further into the future the forecast is. Based on literature (Lorenz, Hurka, Heinemann, & Beyer, 2009), a relative RMSE of 30% for forecasting 24 hours into the future was chosen as a realistic estimate of error. The relative RMSE was calculated by taking the RMSE for forecasts of time steps with non-zero values (time steps during the night were ignored) and scaling that error by the mean value over time.

## Robot Behaviors

Three behaviors are implemented for the agent: monitor, charge, and rest. While the top-level action-selection is handled by the architecture described in section 2, each behavior is itself built as an assemblage of simpler behaviors. In this section each behaviors implementation, activation, and hysteresis will be detailed.

### Monitor Behavior

The monitor behavior is responsible for pest monitoring. It drives the robot around its work space, exploring and searching for pests. This is implemented as a vector summation of three behaviors: a wander behavior to drive the robot around its work space, an attachment behavior to keep the robot in its workspace, and an obstacle avoidance behavior. The wander behavior itself is an assemblage of a random walk behavior, and an avoid-past behavior. This provides a reasonably consistent, but not perfect searching behavior. Nearly every trial had some pests missed during monitoring, meaning the agent had to deal with imperfect measurements of the environment.

### Monitor Activation

For the activation function, the monitor behavior's MOT (equation 4 below) drives the agent to monitor the environment at periodic intervals based on the time since it last monitored, $T_M$. For this experiment, the MOT began increasing after 12 hours since monitoring, and maximized at 168 hours (seven days). The RM (equations 5 and 6) increases activation of the monitor behavior based on the last measured pest level, $P_M$. Two releasing mechanisms are tested for the monitor behavior. The first increases linearly based on $P_M$, scaled so that it maximizes activation at $P_{Max}$ which was set to the pest threshold of 20 in this experiment. The second releasing mechanism increases activation based on a logistic function, the same logistic function the circadian function uses below. The linear RM has a small constant factor to always release the monitor behavior, even if no pests were found, as the environment is always available to be monitored.

$$MOT(\, T_M \,) = \frac{T_M - 12}{156} \tag{4}$$

$$RM(\, P_M) = \frac{\mathrm{P_M}}{\mathrm{P_{Max}}} + 0.01 \tag{5}$$

$$RM(\, P_M) = \frac{1}{1 + \mathrm{e}^{-2(\mathrm{P_M} - 17)}} \tag{6}$$

The CIR (equation 7) increases activation according to the same logistic function as in equation 6, however it uses the forecasted pest level for the current time, $P_F(t)$, rather than the most recent measured level. The reasoning behind this logistic function is easy to see when compared to the linear function in Figure 6. By restricting activation until the pest level is close to the critical threshold, the agent can theoretically save time and energy. However, without measuring the pest population, the agent has no way of knowing for certain when the population is close to this threshold. The intuition behind this circadian function is that forecasts from the past can substitute for current measurements, and allow the agent to more conservatively respond to an increasing level of pests.

$$CIR(\, P_F(t)\,) = \frac{1}{1 + \mathrm{e}^{-2(\mathrm{P_F(t)} - 17)}} \tag{7}$$

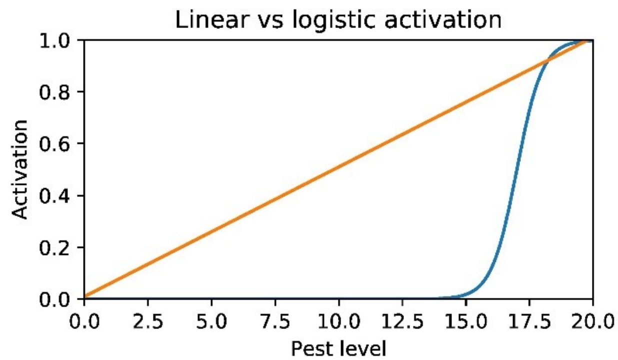


Figure 6. Graph of the logistic and linear activation equations used in the monitor behavior.

### Monitor Hysteresis

The hysteresis for the monitor behavior is a constant thirty minutes, meaning the behavior will execute for at least thirty minutes each time it activates. As the MOT component resets activation when the agent monitors, every monitor execution lasts thirty minutes.

### Charge Behavior

The charge behavior moves the robot to the region of its work space where direct sunlight is available. Once there the agent waits, charging if sunlight is available and the battery is not full, until another behavior takes over executing. This is implemented through a simple finite state automaton (FSA) (see Figure 7).

### Charge Activation

The charge behavior's motivation function (equation 8) is a function of the internal battery level, increasing linearly as the robot's battery level decreases. The RM (equation 9) increases the activation level proportional to the amount of solar energy currently available to the robot. The behavior is released even when there is no sunlight so that the agent can move and wait for energy, rather than monitor when its battery might be too low. In the following equations, B is the battery level of robot (as a percentage), $S_M$ is the current measured solar irradiance, and $S_{Max}$ is the level of solar irradiance where the RM reaches max activation.

$$MOT(\mathrm{B}) = 1 - B \tag{8}$$

$$RM(S_M) = \frac{\mathrm{S_M}}{\mathrm{S_{Max}}} + 0.01 \tag{9}$$

Finally, the circadian function takes the forecasted solar irradiance and estimates three values. First, the energy that would be collected if the agent charged at the current time ($S_{Cur}$). Second, the average energy that would be collected by charging during daylight over the next 24 hours ($S_{Avg}$). Third, the max energy that could be collected by executing the charge behavior once over the next 24 hours ($S_{best}$). The CIR component, defined in equation 10, activates when charging at the current time ($S_{Cur}$) is better than the average ($S_{Avg}$). The activation is proportional to how much better than average charging now is, scaled such that it reaches max activation at the best time to charge (as estimated by $S_{Best}$). In the equations below, $S_F(t)$ is the forecasted solar energy at some point in time, $t$.

$$CIR\big(S_F(t)\big) = \frac{S_{Cur} - S_{Avg}}{S_{Best} - S_{Avg}} \tag{10}$$

The three values $S_{Cur}$, $S_{Avg}$, and $S_{Best}$, are calculated as follows: $S_{Cur}$ sums the forecasted solar irradiance for the next three hours. $S_{Avg}$ estimates the results of $S_{Cur}$ for the next 24 hours, in 30 minutes increments. Values that charged only during daylight are used to find the average, therefore $N$ in equation 12 is the number of time steps with sunlight. $S_{Avg}$ is scaled by a factor $f$, so that values larger or smaller than the average can be used instead. In this work, $f$=1.5. $S_{Best}$ also looks at the results of $S_{Cur}$ over 24 hours, but returns the max value.

$$S_{Cur}(\mathrm{t}) = \sum_{i=t}^{i=t+3} S_F(i) \tag{11}$$

$$S_{Avg}(\mathrm{t}) = \frac{\mathrm{f}}{\mathrm{N}} \sum_{i=t}^{i=t+24} \mathrm{S_{Cur}}(i) \quad if\ S_F(t) > 0\ for\ all\ steps\ in\ S_{Cur}(i) \tag{12}$$

$$S_{Best}(\mathrm{t}) = \max(\{S_{Cur}(t), \dots, S_{Cur}(t+24)\}) \tag{13}$$

### Charge Hysteresis

The charge behavior has a constant hysteresis of three hours, meaning the behavior will stay active for at least three hours once activated. This is enough time to charge a large portion of the agent's battery, assuming it began charging at a good time, and keeps the agent from ending the charge behavior early with a half-filled battery.

### Rest Behavior

The final behavior, rest, simply causes the robot to stop moving. This behavior exists as an alternative to always attempting to charge or monitor, even when the activation of both behaviors is low. Thus, if there is nothing important for the agent to do, the agent will do nothing. The rest behavior maintains a constant activation of 0.5 at all times. It has a brief hysteresis period, ten minutes. There is potential for a more sophisticated activation function for a rest behavior, taking into consideration over heating or bad weather, but this was not the focus of this work. Figure 7 shows the top-level diagram of the robot's architecture for this experiment.

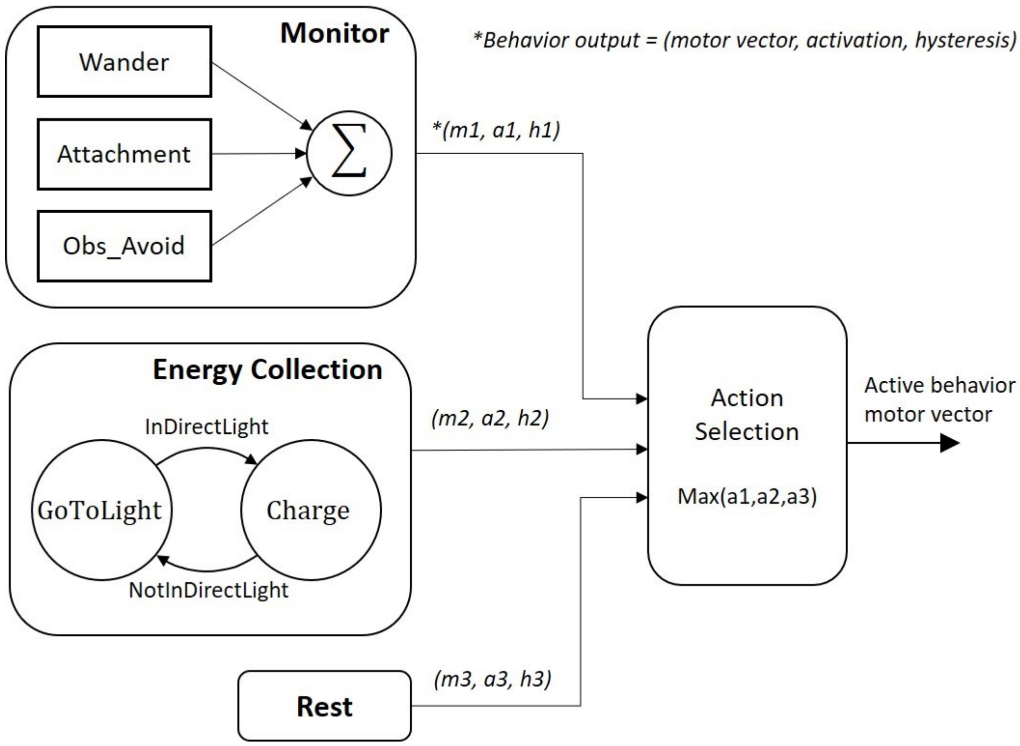


Figure 7. Top-level behavioral architecture

## Experiment Procedure

The robot's task was to detect when the pest population crossed a critical threshold, which was arbitrarily defined as 20 total pests in the robot's workspace. This goal was achieved by executing the monitoring behavior and measuring the pest population after said threshold is reached. Three separate conditions were tested: the inclusion of the circadian function into the monitor behavior with the linear releasing mechanism (monitor-linear), the inclusion of the CIR into the monitor behavior with the logistic releasing mechanism (monitor-logistic), and the inclusion of the CIR into the charge behavior (charge). In each condition, the weight Wc was varied from 0 to 1 by increments of 0.2.

Three metrics are defined. For performance with respect to the goal, the *time-to-detection* is defined as the time between when the critical pest threshold was reached, and when the robot detected it. With respect to energy, *energy-spent* is the average energy used per day by the robot to monitor. *Charge-rate* is the rate of energy collected, per minute, during the periods the robot was charging. These measure the efficiency of both working and charging in terms of energy. Each condition has one associated hypothesis based on these metrics:

Hypothesis 1: For the "monitor-linear" condition, the inclusion of the circadian component will cause the agent to significantly reduce energy-spent without significantly increasing time-to-detection.
Hypothesis 2: For the "monitor-logistic" condition, the inclusion of the circadian component will improve performance, significantly reducing time-to-detection.

Hypothesis 3: For the "charge" condition, the inclusion of the circadian function will significantly increase the charge-rate.

The experiment was executed 20 times for each condition and group. All environmental factors (pest locations, levels, solar irradiance) were saved and replicated, so that equivalent trials were identical between groups. To test whether the performance of groups in one condition are different, a one-way ANOVA was applied, and Tukey's HSD used for post hoc testing. When the assumption of homogeneity was violated, Welch's ANOVA and the Games-Howell post hoc testing were used instead.

## Implementation Details

The simulation is implemented in Gazebo as shown in Figure 8. The robot is solar powered and must spend a significant portion of its time charging. It expends a large amount energy whenever moving, such that the entire battery is depleted after 90 minutes of activity. It expends a small amount of energy while inactive, such that the entire battery is depleted after 7 days. The robot receives "direct" solar light, and is able to charge, when in the charge behavior and near the top wall closest to the "sun" (yellow sphere). The robot's workspace is 16x4 meters, and the agent can move at 5 cm/s.

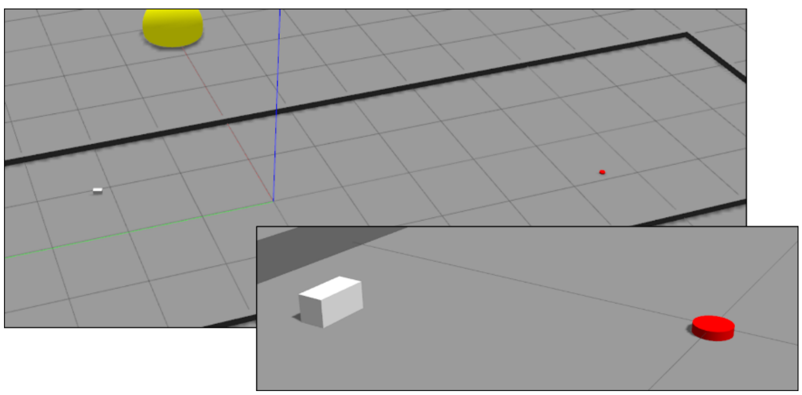

Figure 8. The gazebo simulation environment. The bottom right image highlights the agent (white box) and one pest (red disk).

For all tests, the monitor behavior's weight Wm was set to 0.5, and the charge behavior's weight Wm was set to 0.6. The behavior gains $G$ were left at 1 for all behaviors. When testing a behavior (charge or monitor), the other behavior's circadian function was excluded (Wc=0). When testing the charge behavior, the monitor behavior used the linear releasing mechanism.

# Results and Discussion

## The Monitor Behavior

First let us examine the inclusion of the circadian function for the monitor-linear condition. The data for time-to-detection and energy-spent are shown below in Figure 9. See Appendix A for detailed results of both this and the following conditions. The results for time-to-detection have large variance (a maximum standard deviation of 554 minutes was found for Wc=1.0), as the agent always had the possibility of monitoring right before or after the pest threshold was crossed. No trend, however, emerged in the mean values. A one-way ANOVA found no difference between any groups ( $F(5,114)=0.24$, $p=0.944$ ).

In contrast, the energy-spent shows a clear decreasing trend. With the lowest energy-spent found for Wc=1.0, 24% lower than Wc=0. A one-way ANOVA found the differences between groups was highly significant ($F(5,144)=6.83$ , $p<0.001$). Post hoc testing with Tukey's HSD showed that the energy-spent for Wc=0 was significantly different for groups from Wc=0.4 to Wc=1.0. To summarize, the hypothesis for the monitor-linear condition has been confirmed: Use of the circadian function allowed the agent to significantly reduce its energy expenditure without impacting performance.

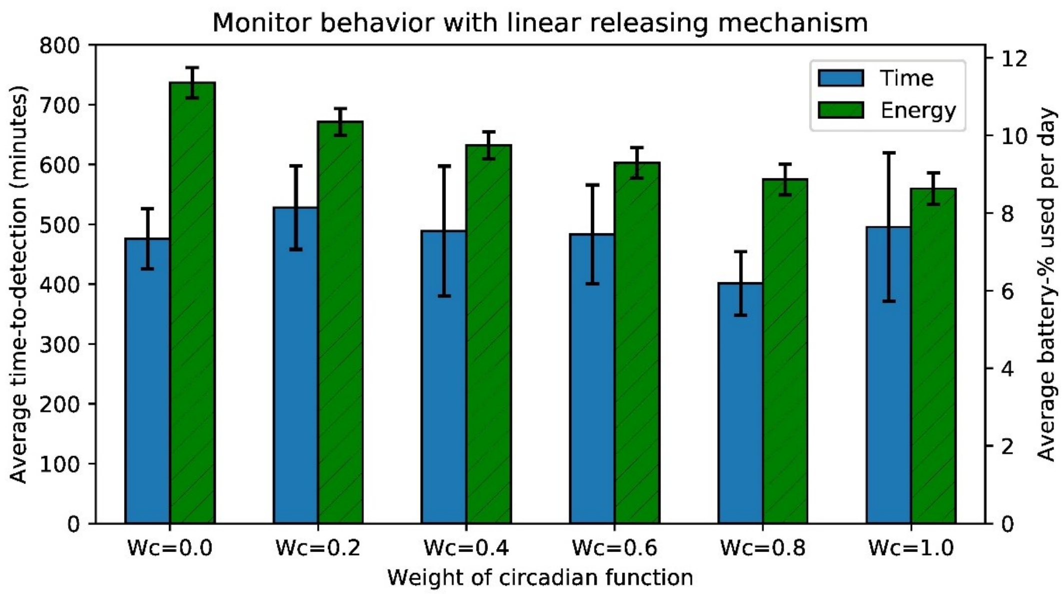


Figure. 9. Average time-to-detection over 20 trials when monitor behavior weights varied. Standard error bars shown.

Next is the monitor-logistic condition, the data is summarized below in Figure 10. In this case, a trend for time-to-detection appears to exist, with the mean value for Wc=1.0 dropping to under half the value for Wc=0. In addition, the variance is larger, with a max standard deviation of 1209 minutes for group Wc=0. For time-to-detection alone, the assumption of homogeneity of variance between groups is violated (Levene's test showed significant difference between variance of groups, $F(5,114)=5.022$, $p<0.001$). Welch's one-way ANOVA was applied and found there was no significant difference between time-to-detection for the groups ($F(5,52.6)=1.285$, $p=0.284$). Therefore hypothesis two could not be confirmed. Perhaps surprising when looking at the data, but the details of the trials shows that the majority of the increase in the average time-to-detection for Wc=0 is due to a minority of very bad trials. Five of the twenty trials have a time-to-detection roughly double (or more) than the average of 1175 minutes. In comparison, there is only one such outlier trial for Wc=1. The circadian function helped the agent be more robust to poor or badly timed measurements by incorporating information from past measurements into its prediction of the pest level.

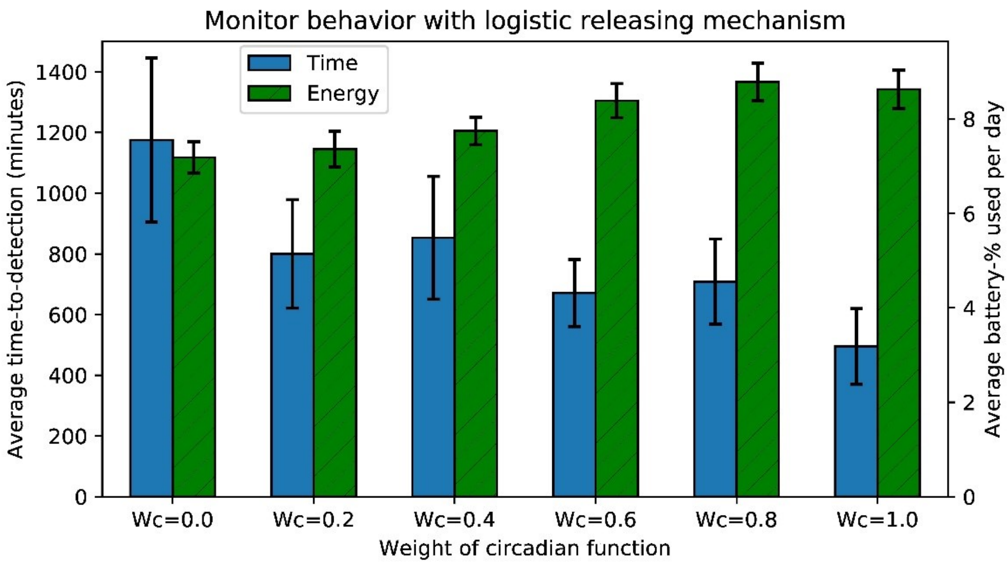


Figure 10. Average energy-spent per day over 20 trials when monitor behavior weights varied. Standard error bars shown.

Although it's impossible in this experimental setup for the agent to know exactly when it needs to monitor in advance, incorporating knowledge about the normal seasonal pest cycle through the circadian function allows the agent to prioritize when it spends time and energy monitoring. Without the circadian system, the agent can achieve similar performance by expending more energy, shown with the monitor-linear condition. However, attempting to use the same conservative activation scheme as the circadian function without the forecasts (as shown in the monitor-logistic condition) caused the agent's performance to become less reliable.

## The Charge Behavior

The results of including the circadian function into the charge behavior are shown below in Figure 11 and 12. The charge rate increases quickly with a small inclusion of circadian function, and seems to level off after. The two horizontal lines of Figure 12 provide a scale for reasonable performance. The bottom line is the average charge rate over all daylight hours, representing the performance of an agent that picks a time to charge during the day randomly. The top line is the average charge rate for the best time to charge every day. Both use the assumption of a three-hour charge time. These can be considered reasonable boundaries for "best" and "worst" performance.

For weight Wc=0.4 and higher, the charge rate was over 10% higher than without the circadian function. A one-way ANOVA found the difference between groups was highly significant ($F(5,114)=17.497$, $p<0.001$). Post hoc testing with Tukey's HSD showed group Wc=0.0 was significantly lower than all other groups, and groups Wc=0.2 to Wc=1.0 were not significantly different from each other. Therefore hypothesis three is confirmed, the agent using the circadian function has significantly higher charge-rate.

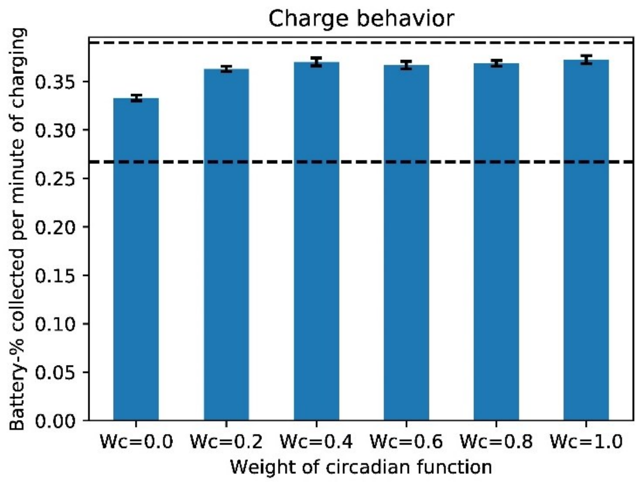


Figure 11. Average charge-rate as the circadian function is included in the charge behavior. The lower dashed line is the average charge rate if the agent charged during all daylight hours. The upper dashed line is the average charge rate if the agent charged at the best time each day. Standard error bars shown.

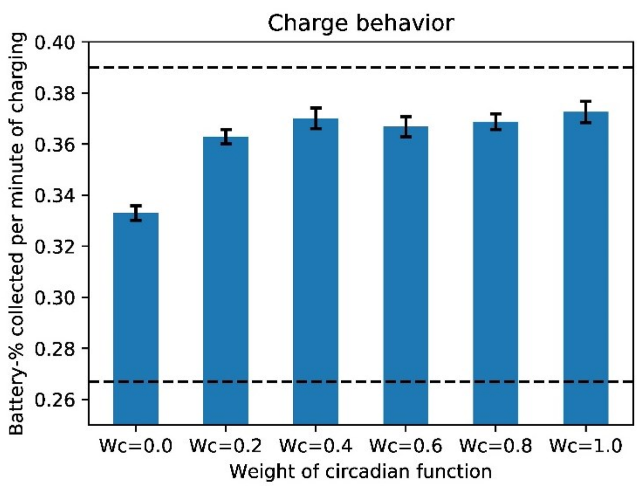


Figure 12. The same data as in figure 11, zoomed in for clarity.

The releasing mechanism for the charge behavior performs reasonably well. The heuristic of "charge when there's a lot of sunlight" is effective. However, it is unable to adapt to changing solar dynamics. Both weather and the seasonal shift in solar irradiance create opportunities for the forecasting approach to perform better. An example of this is shown below in Figure 13, which illustrates a single day that two agents both charged on. For Wc=0, the releasing mechanism prioritizes charging at the highest solar level. The agent with Wc=1.0 starts charging earlier and at a lower level of sunlight, even though it started the day with ~5% more battery power. It does so because it forecasts that future opportunities to charge will be worse, due to inclement weather.

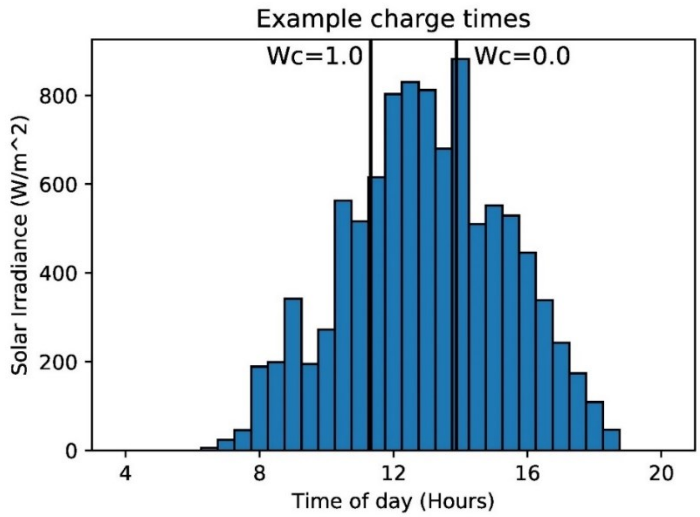


Figure 13. One example when agents charged on the same day highlights how the circadian function allows the agent to identify better times to charge, even when the immediate level of solar irradiance is lower.

Summarizing these results, at Wc=0.6 the ACS allowed the agent to use ~20% less energy and charge ~10% faster. For an energy-constrained agent, this would represent a significant boost in performance, and it is achieved without complete reliance on forecasts. Ensuring the agent could still operate (though with degraded performance) if the time series models became highly inaccurate. For any persistent robotic platform, energy is likely to be a major concern. For solar-powered platforms in particular, it is likely to be the limiting factor in performance.

# Conclusions

This work has detailed an approach to help a robotic agent respond to long-term dynamics - the Artificial Circadian System. Inspired by the circadian systems in nature, time series models forecast the changing environment. These forecasts are incorporated into the action-selection process to allow a robot to act based on both the current measured and future predicted values of the state. Forecasts can potentially be applied in multiple ways: for proactive behavior, to estimate an otherwise unmeasured state, to identify the best time to execute a behavior, or even to detect deviations in the environment by comparing a predicted and measured state.

An agricultural domain was chosen as a test bed, and a pest-monitoring task with a slow, solar-powered robot was simulated. With the ACS, the agent was able to model the pest level over time, even when not actively monitoring it. Allowing the agent to conserve energy while still reliably performing its task. The agent also weighed the future predicted availability of solar energy into its decisions on when to charge, improving the timing of transitions to a charging behavior. The agent's behavior adapted to the changing environment: prioritizing monitoring only when the pest level was high, and prioritizing charging at the best times.

When considering whether the ACS is relevant for a given problem, there are two main factors to consider. One is the predictability or randomness of the relevant environmental dynamics. By relevant, we mean that the state or attribute of the environment has a meaningful impact on the performance of the agent. Purely random change is unpredictable, and there is no better strategy than a reactive one in such a situation. On the opposite end, purely deterministic environments lend themselves to solving an optimal schedule offline. Many environments fall in between these extremes, experiencing random perturbations and short-term trends, but showing clear long-term patterns that can still be exploited. The interaction between the agent and environment is also critical. If the agent does not remain active for long enough to experience meaningful change or is fast enough to react to change with little loss in performance, it may not need to predict anything. If the agent can be considered either slow or persistent, however, then the ability to anticipate change and adapt to it will likely bring benefits.

As shown in the presented experiment, the agricultural domain is an area with significant potential relevance for the ACS. This extends to field robotics in general if the platforms are intended to be persistent. Urban applications may also crop up whenever a robotic agent is impacted by traffic: either pedestrian or vehicular. Traffic dynamics are both highly cyclic and noisy, an ideal case for the modeling approach used in this work. Adapting activity of an agent around traffic using the ACS could allow agents to save time and energy, in the same way that people might avoid driving during rush hour or take an alternative route.

A physical test-bed has been constructed to continue future work in two primary directions. First is to more thoroughly study the scope of this approach. What other characteristics of an environment and agent make the artificial circadian system valuable? How fast does an agent need to be, relative to environmental changes, such that forecasting isn't useful? How accurate do the forecasts need to be? Second is robustness in cases when the forecasting system fails, due to an insufficient model or unpredictable changes in the environment. Any model of a natural environment is limited. How can the agent detect when forecasts accuracy degrades, and adjust the weight of the forecasts in action-selection appropriately? Given detection of the model failing, how should the agent re-entrain it using the state's history?

# Funding

This research is supported by The Office of Naval Research Grant #N00014-15-1-2115.

# References

Ambrus, R., Ekekrantz, J., Folkesson, J., & Jensfelt, P. (2015). Unsupervised learning of spatial-temporal models of objects in a long-term autonomy scenario. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on (pp. 5678-5685). IEEE.

Arkin, R. C. (1998). Behavior-based robotics. MIT press.

Arkin, R. C., & Egerstedt, M. (2015, December). Temporal heterogeneity and the value of slowness in robotic systems. In Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on (pp. 1000-1005). IEEE.

Arkin, R. C., Fujita, M., Takagi, T., & Hasegawa, R. (2003). An ethological and emotional basis for human–robot interaction. Robotics and Autonomous Systems, 42(3-4), 191-201.

Aschoff, J. (1967). Adaptive cycles: their significance for defining environmental hazards. International Journal of Biometeorology, 11(3), 255-278.

Bacher, P., Madsen, H., & Nielsen, H. A. (2009). Online short-term solar power forecasting. Solar Energy, 83(10), 1772-1783.

Blumberg, B. (1994, July). Action-selection in hamsterdam: Lessons from ethology. In Third International Conference on the Simulation of Adaptive Behavior (pp. 108-117).

Boyan, J. A., & Littman, M. L. (2001). Exact solutions to time-dependent MDPs. In Advances in Neural Information Processing Systems (pp. 1026-1032).

Buchli, B., Sutton, F., Beutel, J., & Thiele, L. (2014, November). Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In Proceedings of the 12th ACM conference on embedded network sensor systems (pp. 31-45). ACM.

Chernova, S., & Arkin, R. C. (2007). From deliberative to routine behaviors: a cognitively inspired action-selection mechanism for routine behavior capture. Adaptive Behavior, 15(2), 199-216.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition. Journal of Official Statistics, 6(1), 3-73.

DeCoursey, P. J., Krulas, J. R., Mele, G., & Holley, D. C. (1997). Circadian performance of suprachiasmatic nuclei (SCN)-lesioned antelope ground squirrels in a desert enclosure. Physiology & Behavior, 62(5), 1099-1108.

Fleury, F., Allemand, R., Vavre, F., Fouillet, P., & Bouletreau, M. (2000). Adaptive significance of a circadian clock: temporal segregation of activities reduces intrinsic competitive inferiority in Drosophila parasitoids. Proceedings of the Royal Society of London. Series B: Biological Sciences, 267(1447), 1005-1010.

Hyndman, R., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media.

Hyndman, R. (2017). Forecasting Functions for Time Series and Linear Models. R package. Retrieved from http://pkg.robjhyndman.com/forecast/

Kansal, A., Hsu, J., Zahedi, S., & Srivastava, M. B. (2007). Power management in energy harvesting sensor networks. ACM Transactions on Embedded Computing Systems (TECS), 6(4), 32.

Kindlmann, P., Arditi, R., & Dixon, A. F. G. (2004). A simple aphid population model. In: Aphids in a New Millennium (pp. 325-330).

Krajník, T., Fentanes, J. P., Santos, J. M., & Duckett, T. (2017). Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. IEEE Transactions on Robotics, 33(4), 964-977.

Liu, L., & Sukhatme, G. S. (2018). A Solution to Time-Varying Markov Decision Processes. IEEE Robotics and Automation Letters, 3(3), 1631-1638.

Lorenz, E., Hurka, J., Heinemann, D., & Beyer, H. G. (2009). Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. IEEE Journal of selected topics in applied earth observations and remote sensing, 2(1), 2-10.

National Renewable Energy Laboratory (2016). National Solar Radiation Database. Retrieved from https://maps.nrel.gov/nsrdb-viewer

Mitsou, N. C., & Tzafestas, C. S. (2007, June). Temporal occupancy grid for mobile robot dynamic environment mapping. In 2007 Mediterranean Conference on Control & Automation (pp. 1-8). IEEE.

O'Brien, M. J., & Arkin, R. C. (2017). Modeling Temporally Dynamic Environments for Persistent Autonomous Agents. In Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, (pp. 442-448) The AAAI Press, Palo Alto

Oatley, K. (1974) Circadian Rhythms and Representations of the Environment in Motivational Systems. In: McFarland, D.J. (Eds.), Motivational Control Systems Analysis (pp. 427-459). Oxford, UK: Academic Press.

Paranjpe, D. A., & Sharma, V. (2005). Evolution of temporal order in living organisms. Journal of Circadian Rhythms, 3(1), 7.

Richards, J., & Gumz, M. L. (2013). Mechanism of the circadian clock in physiology. American Journal of Physiology-Regulatory, Integrative and Comparative Physiology, 304(12), R1053-R1064.

Richter, M., Sandamirskaya, Y., & Schöner, G. (2012, October). A robotic architecture for action selection and behavioral organization inspired by human cognition. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 2457-2464). IEEE.

Roenneberg, T., Daan, S., & Merrow, M. (2003). The art of entrainment. Journal of Biological Rhythms, 18(3), 183-194.

Velayudhan, L., & Arkin, R. C. (2017, December). Sloth and slow loris inspired behavioral controller for a robotic agent. In Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on (pp. 1880-1885). IEEE.

The Robotics Industry Will Reach $237 Billion in Revenue Worldwide by 2022. (2018) Tratica. Retrieved from https://www.tractica.com/newsroom/press-releases/the-robotics-industry-will-reach-237-billion-in-revenue-worldwide-by-2022/

# Appendix A: Experimental Data

In this section, we present the experimental results in tabular form for reference.

Table 1. Detailed results for monitor-linear condition. Time-to-detection is in minutes, and energy-spent in battery percentage per day.

| Weight of CIR | | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|---|
| Time-to-detection | Mean | 475.85 | 527.65 | 488.45 | 483.25 | 401.05 | 495.5 |
| | Stand. Dev. | 224.84 | 311.13 | 485.02 | 369.82 | 237.15 | 554.61 |
| Energy-Spent | Mean | 11.206 | 10.206 | 9.618 | 9.1641 | 8.7467 | 8.5187 |
| | Stand. Dev. | 1.7783 | 1.5457 | 1.5657 | 1.7512 | 1.783 | 1.8513 |

Table 2. Detailed results for monitor-logistic condition. Time-to-detection is in minutes, and energy-spent in battery percentage per day.

| Weight of CIR | | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|---|
| Time-to-detection | Mean | 1175 | 800.1 | 853.15 | 671.4 | 708.6 | 495.5 |
| | Stand. Dev. | 1208.8 | 798.8 | 905.1 | 494.1 | 629.02 | 554.61 |
| Energy-Spent | Mean | 7.0889 | 7.268 | 7.6443 | 8.2783 | 8.6715 | 8.5187 |
| | Stand. Dev. | 1.4616 | 1.7046 | 1.3049 | 1.6217 | 1.7935 | 1.8513 |

Table 3. Detailed results for charge condition. Charge-rate is in battery percentage per minute.

| Weight of CIR | | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|---|
| Charge -rate | Mean | 0.33299 | 0.3629 | 0.37007 | 0.36684 | 0.36875 | 0.3726 |
| | Stand. Dev. | 0.000163 | 0.000153 | 0.00033 | 0.000309 | 0.000188 | 0.00035 |