Qualitative Prediction-Observation: Using Robotic Surprise for Learning by Experimentation

Alex Juarez, Ashok Mohan, Timo Henne, Erwin Prassler

Univ. of Applied Sciences Bonn-Rhein-Sieg, Dept. of Computer Science
Grantham Allee 20, 53757 Sankt Augustin, Germany
{alex.juarez, ashok.mohan, timo.henne, erwin.prassler} 'at' fh-brs.de

***Abstract*— The ability of performing open-ended learning is a key feature of natural cognitive systems. One approach to achieve this ability is to enable an agent to learn by experimentation. For this, the agent is bound to have a model of the physical world based on a set of hypotheses that allow to predict the future state of the world, or the result of certain actions. Significant deviations or contradictions between predicted and observed results can be used to refute the model and trigger a revision of the hypotheses. For detecting these deviations we propose a mechanism called *QPOLE* which stands for Qualitative Prediction-Observation Loop for Learning by Experimentation. The *prediction module* of this loop uses a qualitative model of the world to predict temporal states of the system. An *observation module* performs an online comparison of this prediction and the observed numerical data. The data used by the observation engine is supplied by a visual sensor which monitors the environment and thus captures the results of the agents actions. We present and discuss the results of first experiments with QPOLE for simple real-world experiments with a rolling and bouncing ball.**

I. INTRODUCTION

The capability to learn and acquire new knowledge is most fundamental for any creature which is exposed to a changing, possibly hostile environment. A creature which is not able to learn proper responses to events and changes in the environment may not survive. [1] Given that this trivial insight does not only hold for any natural forms of life but also for artificial creatures, a vast amount of work has been devoted in the past four decades to study learning mechanisms and develop computational models and learning paradigms. The overall objective of this work is to enable such artificial creatures to acquire new knowledge from their environment and adapt to and account for changes and challenges which it poses.

Significantly less work has been invested in studying and modeling the processes which initiate and trigger learning, although this is everything but an issue of lesser importance for the design of truly autonomous and intelligent agents. For some forms or learning (*supervised learning*) the effect of triggering learning does not come from the individual itself, but from a teacher. In our work we are looking into a different form of learning, which does not involve a teacher (*unsupervised learning*). In order to achieve open-ended learning within this paradigm, we focussed our work on *robotic learning and discovery by experimentation* [1]. We are investigating and developing mechanisms which enable an intelligent autonomous robot to gain new insights about the surrounding world and physical phenomena by conducting experiments with objects which it finds in the real world. As an example, we want a robot to discover the physical laws affecting the motion of a ball rolling in the plane and eventually bouncing against a wall. Another example is to let the robot discover the concept of *articulation* by experimenting with objects which are connected by joints.

In such a context, the question of what motivates and triggers learning is a very important one. Why and when should a robot learn if it does not have any innate mechanisms which initiate the learning process? Of course, one can say that a robot should always learn once it is turned on, but that would not be an appropriate answer to the question. In natural agents learning is typically initiated by some internal drives or by an external stimulus. It seems to be advisable to take some inspiration from nature when thinking about mechanisms which may cause a robot to learn. One might, for example, equip the robot with some form of artificial curiosity [2]. We propose a computational approach for another driving force for learning, namely *robotic surprise*.

Assume that our robot has already developed some knowledge about rolling balls and the physics behind rolling balls. If the robot perceives a ball that rolls on a plane, it will understand and be able to explain what is going on. Now assume that the ball bounces against a wall and the robot has never seen this before. The ball does something *unexpected*, which the robot is no longer able to explain. We call this a *surprise* and use it to initiate learning by experimentation.

The major scientific contributions of this paper are the following: We propose a computational approach to *artificial surprise* which we call QPOLE (Qualitative Prediction-Observation Loop for Learning by Experimentation). This approach is not a theoretical one as for example in [3], [4] but faces the problems which an embodied agent has to confront. It operates on real sensor data, which are noisy and sometimes give only partial insights into the phenomena going on in the environment. Our approach is able to match observations which provide only a limited number of features with models of physical phenomena involving a superset of

The work described in this article has been partially funded by the European Commission's Sixth Framework Programme under contract no. 029427 as part of the Specific Targeted Research Project XPERO ("Robotic Learning by Experimentation")

[1]This, of course, holds even more so for entire species, but we do not address on the issue of evolutionary and genetic learning in this work.

those features. The approach performs an *online* comparison of predictions based on qualitative models with quantitative and noisy sensor data, while in other work [5] this is done *offline*. The quantitative sensor data are converted into a qualitative description by temporal abstraction, which is also able to handle spurious outliers in the sensor data.

The paper is organized in five sections. After this introduction we briefly discuss some related work in model building and revision, which is at the core of our work. In Section III we present QPOLE, our approach to matching of observations and predictions of physical phenomena such as the rolling and bouncing of a ball against a wall. This presentation is followed by a detailed discussion of some first simple experiments to investigate the performance of the approach. In the final section we draw some conclusions and give an outlook on our future work.

II. Related work

The creation and revision of models that explain and predict the behavior of systems and processes, has been studied by disciplines such as system dynamics, control theory and statistics. Some of the most fundamental methods in this area are *model validation, system identification* and *qualitative modeling*.

Model validation is the application of methods and techniques that substantiate the accuracy and consistency of a model with respect to its application to real world situations. Some of the techniques used in model validation are: comparison to other models, degenerate tests, event validity tests, multistage validation and predictive validation [6], [7].

System identification is the process of combining a partially-specified model of a system with observations of its behavior and revising it until it converges on a more accurate and precise model. Among the techniques used to estimate the parameters of the model are standard least squares distance or the prediction error criterion. When only sparse knowledge of the system is available, closed-loop variants propose an iterative adaptation of the model parameters, using the information obtained by the application of identification techniques to input data samples [8], [9].

Qualitative models have proved to be extremely efficient in representing incomplete knowledge. Furthermore, qualitative descriptions are able to capture the distinctive characteristics of the systems they describe, while ignoring their quantitative details [10]. Qualitative modeling consists of creating a model of a system by means of qualitative descriptions. These descriptions are capable of representing incomplete knowledge of the structure and behavior of the system, behaviors that can be predicted from the model which consists of the parameters, its domains, various constraints on the parameters and an initial state definition of the system[11], [12], [13].

Most notably, the approach presented in [14] investigates the modeling of dynamic systems by means of qualitative simulation, in order to diagnose system faults. This approach creates fault models based on observations of the system behavior, and then combines them with inductive learning to produce hypotheses on the system faults. These hypotheses are then transformed into qualitative models (predictions) which are verified against new observation, by means of a qualitative matching mechanism. This mechanism presents similar ideas to our approach, however, it does *not* address the issues associated with its application to a real robot that learns by experimentation, such as the embodiment, environment and sensor data characteristics.

Semi-quantitative modeling methods such as the ones proposed by Q2 [15] and its successor Q3 [16] allow for the use of some available quantitative information. Although these quantitative information is not enough to perform pure numeric simulation, it greatly aids in making the qualitative prediction more specific and tractable. Both qualitative and semi-quantitative modeling and simulation have also been combined with system identification and model validation methods. An example is SQUID [5], a system identification method which attempts to match semi-quantitative trends to semi-quantitative behaviors. For this, the space of potential models is defined by semi-quantitative differential equations.

In robotics, the prediction and observation of events has recently obtained attention from the research community. Lee et al. [17] present an approach that models the sensory perception of a computer simulated learning agent. This model is used to predict the behavior of the agent and to validate that the simulation created is more accurate than those that can be obtained by classical simulation methods.

Cognitively inspired approaches propose curiosity and surprise mechanisms that detect divergence between expected and perceived events. The intensity of the divergence is given, for example, by the probability of not expecting a perceived object or event at different levels of abstraction, given that they are stored in memory [3], [4]. When applied to a real environment, however, these approaches often face the problem of employing models that are too simple for describing the underlying observed phenomena or the complexity of the environment where the robot exists.

III. Prediction-Observation-Surprise

A. The Robot and its Surrounding World

The environment used to test QPOLE is a small rectangular arena of 1.20 x 0.80 meter side-length. The arena is surrounded by four white walls, 20 cm in height. We assume that the robot performs some assigned task such as moving around in the arena. The robot we used for our experiments is a Kephera II. A color CMOS camera with a resolution of 640x480 pixels and a framerate of approx. 30 fps is mounted on top of the robot pointing in the direction of the robot's orientation. The robot uses this camera to perceive the surrounding environment. Additionally, we also assume that the robot knows about its own position and those of other objects in the environment with respect to a predefined reference frame.

In our experiments only a single object (a yellow ball) was placed in the arena. We assume that the robot is able to identify the yellow ball as a known object. The robot has some limited knowledge about rolling balls, which is

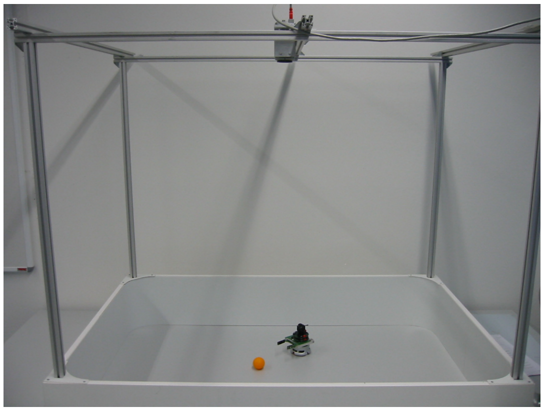

Fig. 1. The environment in which the robot moves and acts.

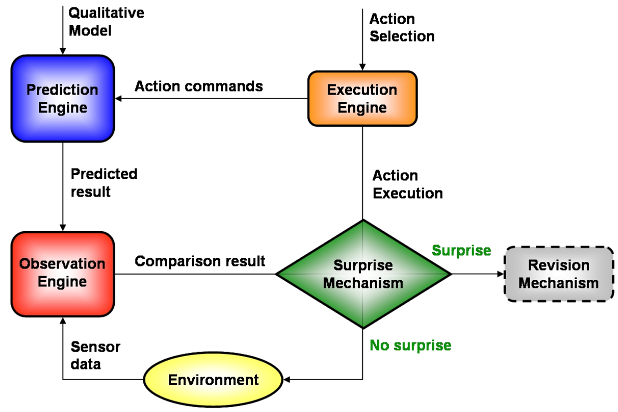

Fig. 2. The data and process flow of QPOLE.

represented by the qualitative model in Experiment 1 in Section IV.

In the subsequent sections we are going to describe the details of our approach QPOLE, and of how the robot is put into a state of surprise which should cause it to start learning by experimenting with the yellow ball and find out more about the physics of rolling and bouncing balls.

B. Data and Process Flow in QPOLE

The data and process flow in QPOLE is shown in Figure 2. QPOLE uses qualitative models to predict the underlying phenomena resulting from the execution of a set of actions that are known and available to the robot, e.g. pushing an object. These models are created by a *prediction module* by means of qualitative simulation, and then stored in an internal memory. The data gathered by the robot during the actual execution of an action is abstracted to qualitative values (e.g. increasing, decreasing, steady) by an *observation module*. This is an important process in our approach since it allows to compare the sensor data (quantitative by nature) with the qualitative prediction.

The comparison to the qualitative prediction available for the action yields a measure of match or mismatch. In the case of a sufficiently accurate prediction, no mismatch is detected. The model is confirmed and the execution of scheduled actions, e.g. moving around, continues. In the case of a significant deviation, a *surprise module* stops the scheduled course of actions and enters what we call an experimental loop [1] to analyze the phenomenon which caused the surprise and revise the invalid model or generate a new one.

C. Prediction

Qualitative simulation [10], [11] predicts all possible behaviors of a physical system based on the model of the system.

Before the robot starts executing an action or action sequence, the prediction module of QPOLE uses the simulation software QSIM [11] to perform simulation of a *qualitative model*, which is either hand-coded or generated by some learning mechanism. QSIM generates a behavior tree of states starting from a pre-specified initial state. The most relevant approaches to qualitative modeling and simulation are briefly discussed in the following paragraphs.

1) Qualitative models: Ordinary Differential Equations (ODEs) may seem appropriate for modelling the real world, but they are inherently weak in expressing incomplete knowledge. Qualitative models on the other hand are capable of expressing incomplete knowledge of the world and in spite of the incompleteness of the model they are able to derive a complete set of possible behaviors.

Qualitative Differential Equations (QDEs) provide an efficient way to define a qualitative model. A QDE model is an abstraction of an ODE. It consists of real-valued variables, functional, algebraic and differential constraints. The values of the variables in a QDE are described in terms of their ordinal relations and not in terms of real numbers. Similarly the functional relations between the variables are described as monotonic functions and not by an actual function. [11], [18]

2) QSIM: The simulation software QSIM provides the representations and algorithms necessary for qualitative simulation. The structure required to facilitate qualitative simulation includes a qualitative description of the range of each variable and the qualitative relationships between variables. The steps required for Qualitative Simulation using QSIM are as follows:

1) Define the structure of a mechanism as one or more qualitative differential equations (QDEs) in a specific format that QSIM can parse.
2) Define the functions that enable transitions between QDEs if there is more than one QDE.
3) Specify the initial conditions (state) for the simulation.
4) Generate a behavior tree for the model created above using QSIM.

Figure 3 shows the prediction process. It starts from a pre-specified initial state and generates a state tree of behaviors with qualitative values associated with each state. The prediction runs until all the variables have reached a steady state or until a pre defined number of levels. Every path from the root of the behavior tree down to a leaf node represents a possible qualitative behavior [11], [18].

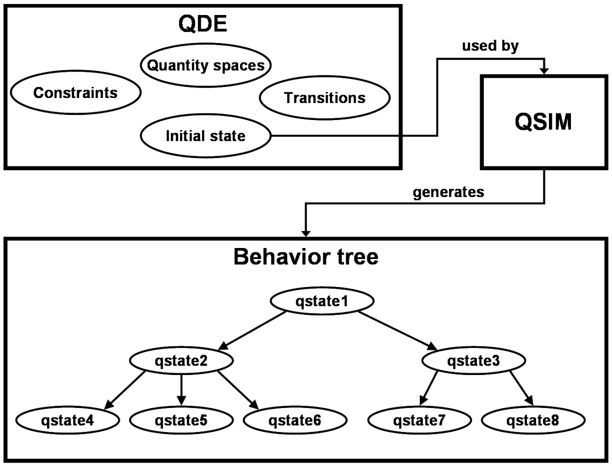

Fig. 3. The prediction process

The behavior tree shows the possible temporal states achievable by the defined model. The nodes are the states of the model while the links of the tree are transitions from one state to another. The nodes contain the qualitative values of each variable defined in the model. For example, in the behavior tree corresponding to the model of a ball rolling, the initial state of the system will transcribe to the root node of the tree, which would specify the velocity of the ball as the maximum observable. The immediate child node(s) of the root node would represent any of the possible temporal states reachable from this initial state, e.g. a velocity between maximum and zero. The next and possibly the last node in the tree may correspond to the state of the system where the velocity reaches zero.

The behavior tree is then sent to the observation module which compares these temporal states to the observed data.

D. Observation

The robot uses its onboard sensors to perceive the surrounding environment in which it moves and acts. The data coming from these sensors are by nature quantitative, not qualitative. The *observation module* of QPOLE processes these data and transforms them to qualitative values which can be compared to the ones predicted by the qualitative model. The robot's primary sensor is an onboard camera, whose images are processed by computer vision routines. The information which is needed and used in the experiments described below, is *position, velocity, acceleration* and also *approximate perceived size of the object* at a specific time instance. We use color segmentation and the CAMSHIFT algorithm [19] to track the color and motion of an object.

The decision on which variables are to be observed and transformed into qualitative values, is based on the prediction model and a predefined list of variables that are observable by the robot. For example, if the current prediction corresponds to the experiment "bouncing ball" (lift the ball and then drop it), the variables present in the model are mass, perceived size, position, velocity, acceleration. However, the observable variables (from the available sensors) are only perceived size, position, velocity and acceleration. Only these can be used in the prediction and observation loop.

The transformation from quantitative to qualitative values as the robot executes an action, is performed by a *temporal abstraction mechanism.* Temporal abstraction is a set of techniques that extracts information from a large amount of raw data to a new abstract representation. Such techniques may include trend templates, temporal interpolation and temporal inference among others [20].

Consider, for example, the case of a ball rolling on the floor, colliding with a wall and bouncing back. An observation of such an event will typically consist of a sequence of positions with time-stamps. Temporal abstraction techniques allow to fit curvature segments to these raw data and then provide a more abstract representation such as *moving forward - colliding - moving backward*, or *increasing velocity - collision - decreasing velocity*. Similar mechanisms have been successfully applied to various domains such as medical information analysis, hierarchical and abstract planning and reinforcement learning. These applications work mostly with *offline datasets*, e.g. monitoring of blood glucose in diabetic patients from controls of past months [21].

In this paper, we propose a qualitative-quantitative comparison method that extracts trends from online data coming from the robot sensors. Such data can be interpreted as a signal that contains the desired qualitative information (trends). For example, the perceived size of the ball that rolls away from the robot will decrease with each time step, thus, the "signal" that corresponds to the perceived ball size will show a decreasing trend. This circumstance allows us to incorporate techniques from time series analysis such as a weighted moving average that allows a reconstruction of measured signals. By this we can obtain the trends based on the current and a relatively small set of previous measurements (approximation window). Additionally, it offers the advantage of a better approximation of the original signal with greater flexibility at defining the window size. This is expressed in Equation 1

$$Y_n = \frac{C_1 X_1 + C_2 X_2 + ... + C_n X_n}{n} \tag{1}$$

where $\{C_1, C_2, ..., C_n\} \in [0..1]\ and\ \sum_{n=1}^{N} C_n = 1$ with a relation that is proportionally inverse to time. This allows to increase the influence of past inputs with respect to new ones in the signal reconstruction. The trend corresponds to the orientation of the reconstructed signal in the approximation window.

To reduce the effects of noise in the trend extraction, hysteresis is applied to the smoothed signal obtained by the moving average technique. The implementation of hysteresis as a signal delay follows Equation 2

$$y(t) = x(at - b) \tag{2}$$

where $a \in \mathbb{R}\ and\ b \in \mathbb{N}^+$. The response delay obtained by setting $a = 1$ is activated by a change in the orientation detected on the smoothed signal. In this way, only if the change in orientation is constant for a period $t > b$, the trend is considered to have changed.

E. Surprise

QPOLE takes the output of the prediction and observation modules, namely the qualitative model and the trends extracted from sensor data, and compares them. If a mismatch is detected, the current model is considered as refuted by the current observation. The mismatch between observation and prediction is what we denote as *surprise*. The surprise and the circumstances under which it occured (e.g. action, time, feature preconditions etc.) can then be used by another (external) mechanism for eventually revising the prediction model, or for repeating the last action or action sequence, thus closing the prediction-observation loop.

The process of comparison of the qualitative model to the extracted trends is called *trend matching*. For each variable suitable for comparison (i.e. variables that are in the qualitative model and are observable), the current trend received from the observation module is compared to the corresponding qualitative state (e.g. increasing, decreasing or constant trends) for this variable in the behavior tree. The comparison starts with the initial state and follows the sequence of predicted states in an ordered way.

If a change in the observed trend is detected (e.g. a change from increasing to constant value in the variable "perceived-ball-size"), the next node in the behavior tree is examined. If there is a match between the new observed trend and the next qualitative state in the behavior tree, the observation continues. Otherwise, it is concluded that the model does not predict the observation accurately, and a *surprise* signal is triggered, which also stops the execution of the current and scheduled actions.

If on the other hand all the nodes in the behavior tree of the qualitative prediction have been visited and no surprise has been triggered, the model is considered accurate. The observation stops and the robot prepares to execute the next scheduled action or action sequence. If in the robot's memory there exists a model for such action, it is used. Otherwise a new model must be created, entering the prediction-observation loop again.

IV. Experiments

In this section we present three experiments that were used for initial testing of our approach. In the first one, the robot pushes the ball which comes to a stop after rolling for some time, owing to friction and air resistance. The second experiment is similar to the first one, but here an obstacle (a wall) is introduced in the path of the rolling ball. This makes the ball rebound towards the robot and thus causes a deviation from the expected behavior. In the third experiment, the robot lifts the ball and drops it, thus causing it to bounce vertically.

A. Experiment 1: Rolling ball.

In this experiment the robot pushes the ball which starts to roll until it stops naturally. From the point of view of the robot, the directly observable features are the color and the perceived size of the ball. We use the color to identify the ball. The measured parameter is the size which varies with time.

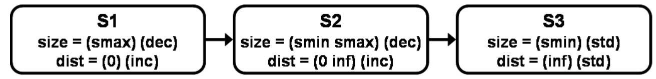

Fig. 4. Experiment 1: The behavior tree generated by the prediction module for a ball rolling and stopping due to friction, with the variables `size` of the ball and distance `dist` between robot and ball. The terms in brackets behind each variable specify the qualitative value and the qualitative trend in each state.

The prediction module generates the behavior tree shown in Figure 4, where the variable values in parentheses stand for the following qualitative values:

- *smax*: initially perceived size of the ball
- *smin*: any perceived size between zero and smax
- *inf*: infinite value
- *inc / dec / std*: increasing/decreasing/steady value

In the case of two values in parentheses (like in state S2), this indicates that the variable in this state will have a value in between the two given qualitative values.

In this experiment the prediction generates 3 consecutive states: the initial state S1 and the predicted states S2 and S3. As expected, the prediction shows the ball rolling (state S2) and then coming to a halt because of air resistance and friction (state S3). The frictional component is modeled as part of the system involving the ball. If the model did not include the frictional component then the prediction would be a ball rolling forever. This indicates that the experimenter needs to decide on how specific the model must be. The more specific the model, the closer the prediction will be to reality.

The observation of the experiment starts the instant before the robot makes contact with the ball. The perceptual information extracted is the size (area) of the color blob associated with the ball (perceived ball size). Figure 5 shows the variation in object size with respect to time taken across approx. 5 seconds of observation of the event.

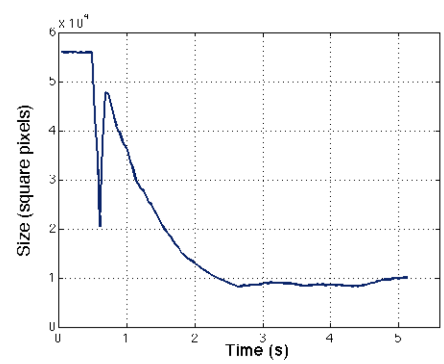

Fig. 5. Experiment 1: Observed area (in squared pixels) of the color blob that represents the ball that the robot sees. The observation starts the instant before the robot pushes the ball ($t = 0.3$). The observed data shows a decreasing trend where effects of noise can be observed at $t = 0.6$. The observation ends when the ball has stopped and all the nodes in the predicted behavior tree have been visited ($t = 5.2$).

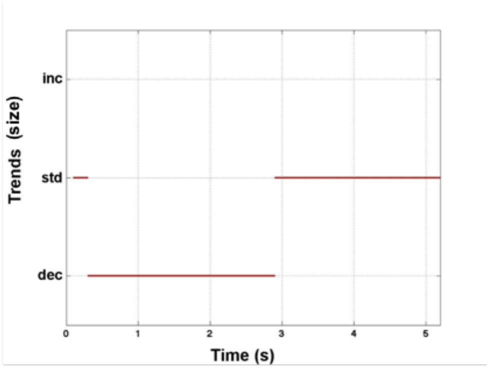

Fig. 6. Experiment 1: Visualization of the trends extracted using the time abstraction mechanism for Experiment 1. The trends are consistent with the numerical observations: an initial steady value is observed in the instant before the robot pushes the ball (std), followed by a decreasing trend of the area value as the ball rolls away (dec). When the ball stops, the trend changes back to steady (std).

The temporal abstraction mechanism extracts two trends from the data: first a decreasing trend from $t = 0.3$ until $t = 2.8$, that changes into a constant (steady) trend from t=2.9 onwards, as shown in Figure 6. The hysteresis applied to the smoothed signal (a delay of 15 measurements to detect a change in trend), allows to ignore the effects of noise in the data. This is depicted again in Figure 5 at $t = 0.6$ where a noisy measurement causes a drop in the measurement of the perceived ball size, and the mechanism successfully ignores it.

This experiment shows how the trends extracted from the observed data were successfully matched to the qualitative prediction. The spurious noise in the observation ($t = 0.6$) was handled correctly by the temporal abstraction mechanism. The phenomena was modeled using two variables: approximate perceived ball size and distance travelled by the ball. Due to the embodiment characteristics, however, the only observable variable is the approximate perceived ball size.

B. *Experiment 2: Rolling ball bouncing off the wall.*

This experiment tests the case when surprise is elicited due to the use of a model that can not explain an observation. For this purpose, the model created for Experiment 1 was used to try to explain the behavior of a ball that is pushed by the robot, rolls away from it, collides with an obstacle (a wall) and bounces back, rolling towards the robot.

The observed variable is, again, the perceived ball size, and its measured behavior is shown in Figure 7. As predicted, the observed size of the ball starts decreasing until it collides. After that, however, it increases as the ball approaches the robot again. As soon as this deviation is encountered, QPOLE triggers a surprise. This can be used as an indication for a need of performing the action again (if possible) to confirm the deviation, or to design another experiment that tests the validity of the model. The result of these external mechanisms could also be used to refine the model so that future observations may concur with it.

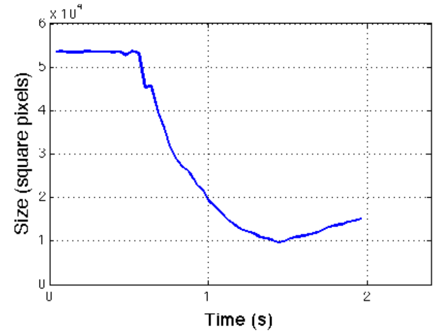

Fig. 7. Experiment 2: Observed area (in squared pixels) of the color blob that represents the ball that the robot sees. The observation starts the instant before the robot pushes the ball. The ball rolls until it bounces off the wall. The observation stops when the perceived data does not match the predicted behavior (time $t = 1.9$).

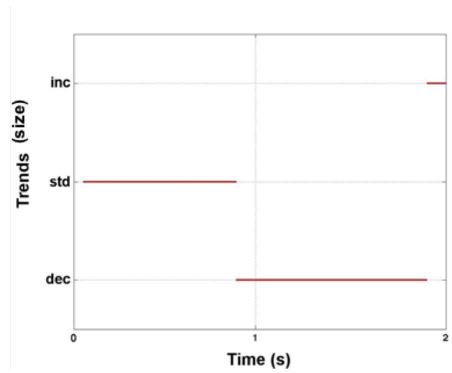

Fig. 8. Experiment 2: Visualization of the trends extracted using the time abstraction. In our experiment, we used the model built for Experiment 1, resulting in surprise triggered at time $t = 1.9$ where the size (area) trend changes from decreasing to increasing while predicted trend changes from decreasing to steady).

This experiment demonstrates the case of a mismatch between the prediction and observation. The prediction created for Experiment 1 was used to explain a rolling ball that collides with a wall and bounces back towards the robot. The observed variable is again the approximate perceived ball size. The trends extracted (see Figure 8) show a decreasing trend followed by a change to an increasing one. Such change triggers a surprise given that the predicted trend sequence was decreasing-constant (see states S2-S3 in Figure 4). In this case a new qualitative model for this phenomena must be created.

C. *Experiment 3: Dropping ball.*

In Experiment 2 (rolling ball bouncing back), surprise is triggered when the observed phenomena can not be explained by a model that was originally designed for Experiment 1 (rolling ball stopping naturally after some time). In this case, the model was correct for one kind of phenomena, but incorrect for another.

However, it is also possible that the qualitative model is correct for the ideal case, i.e. explains phenomena correctly under ideal conditions, but fails to explain it in a real world environment. This is the case depicted by Experiment 3, where the qualitative model correctly predicts what happens to a falling ball in ideal conditions, namely, it bounces in the same place each time to a lesser height until it comes to a stop.When observing this phenomenon, however, the ball does not bounce in the same place, but it moves slightly sideways between bounces.

To be able to model this phenomena, it is needed to gather and introduce into the model data concerning the causes of such "noise", e.g. the inclination of the ground and the elastic properties of the ball. This is a limitation of qualitative modeling: if no detailed information about the phenomena and environment exists when building the model, the prediction will be an approximate or ideal case of what can actually be observed.

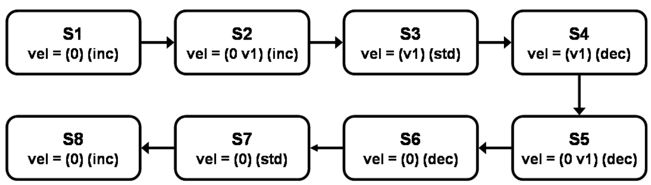

Fig. 9. Experiment 3: The behavior tree generated by the prediction module for a ball bouncing, with `vel` as the variable for velocity. State S1 , the initial state depicts the point at which the ball is dropped. Velocity at this point is zero and increasing. The next state S2 depicts the ball at some point between the initial point and the floor with a positive velocity which is still increasing. S3 is a particularly interesting state as it depicts the point of bounce. The point to note here is that this particular state is observable only for a fraction of a second. State S4 depicts the beginning of the ascent of the ball after the bounce. The velocity begins to decrease at this state through S5 and S6. Stage S7 is also interesting as it depicts the instant in time at which the ball reaches the highest point of its first bounce. From state S8 onwards the cycle repeats itself.

The model is created for a system assuming no air resistance or surface friction. One reason for this is that the damping of the bounce is something that the robot must learn by refining the model, using semi-quantitative description which is not yet realized within QPOLE. The prediction module generates a behavior tree like the one shown in Figure 9. The expected behavior of the system, according to such prediction is the ball bouncing vertically at the same location with a change in the direction of velocity in one dimension. This behavior is the one that is compared online by QPOLE with the observation made in this experiment.

The observable variables (also present in the qualitative model) are the perceived size of the ball, represented by the area of the color blob, the displacement in two dimensions (relative to the image plane) and the computed velocity of the ball as it bounces. Both variables trigger surprise as they deviate from the predicted behavior (Figure 10).

The size of the ball is predicted to remain constant over the observation. However, the perceptual information reveals that the size of the ball does change when dropping a ball. Possible causes for this observation are the material of the ball or a slightly tilted collision surface.

The velocity of the ball was measured by taking the center of mass of the color blob associated to it and tracking it during the observation. The measure is given in pixels per second and represents the displacement in pixels that the robot is able to "see". The deviation from the expected behavior is partly due to the nature of the qualitative prediction.

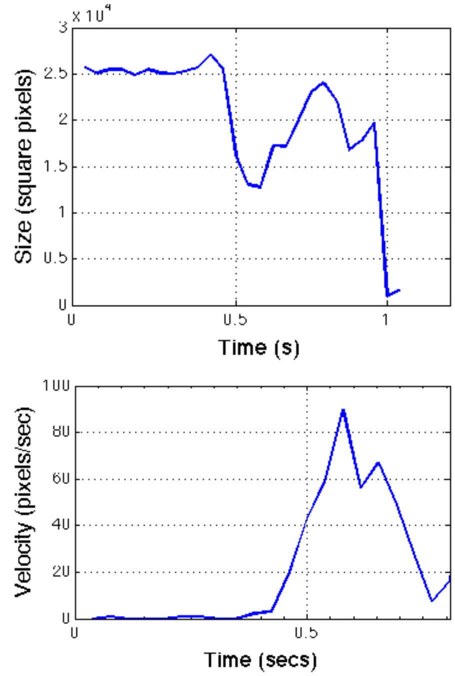

Fig. 10. Experiment 3: Observed variables in this experiment are the perceived ball size and the velocity of the ball as it falls. The data shows unstable perceived size of the ball (top graphic) caused by the ball bouncing away on the floor. The measured velocity (bottom graphic) shows one bounce of the ball (between $t = 0.55$ and $t = 0.8$).

Such prediction states that a steady (constant) velocity is reached at each contact with the floor and at each return to original drop position.

The observation of such a state proved to be very difficult, especially in cases where the "steadiness" lasted only one or two of frames or even less. In our experiments, the prediction was correctly matched when the steady behavior of the model was temporarily skipped when looking for the next node in the behavior tree. Figure 11 shows the trends obtained before surprise was triggered.

The results obtained from this experiment show a limitation of the qualitative model used in the third experiment to generate the prediction. The model is correct for phenomena observed under ideal conditions, e.g. a ball bouncing in the same place. However, the consistent noise in the data, due to the physical characteristics of a real environment and its perceptions, triggers surprise. This indicates that the model must be revised and modified to account for such noisy conditions. One approach to do that is in fact also to refer to semi-quantitative models as for example discussed in [5].

V. Discussion

The current qualitative representation used in our approach appears well suited for the kind of problems that have been contemplated in this paper. However, a more powerful mechanism may be needed when analyzing experiments that involve asynchronous observation of variables. Also, the introduction of semi-quantitative differential equations will be necessary to improve the accuracy of the world model

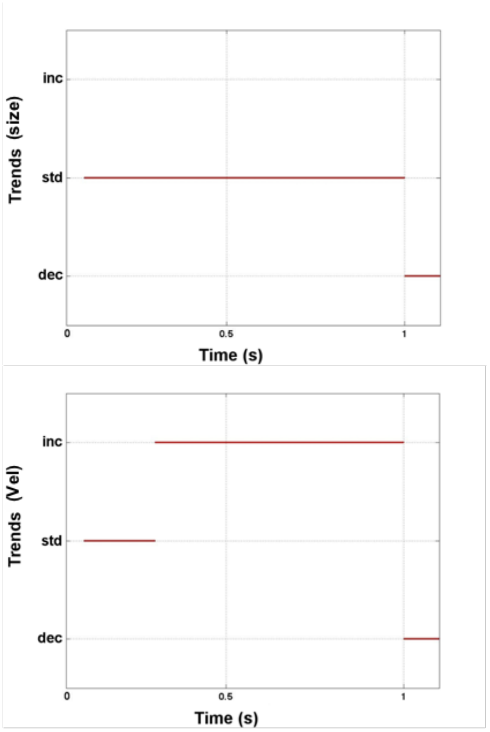

Fig. 11. Experiment 3: Visualization of the trends extracted using the time abstraction. In our experiment, the model used predicted a constant value for the variable size as well as a sequence of steady, increasing and decreasing values for velocity. This resulted in surprise being triggered at time $t = 1.2$ (top graphic) for the perceived size of the ball, when the changing trends in the original data can not be considered as noise anymore. Surprise was also triggered for the velocity variable at time $t = 0.5$ (bottom graphic), when a predicted steady state could not be observed.

or include phenomena such as friction in more complex experiments like "dropping a ball".

The surprises triggered by QPOLE can be passed on to external mechanisms which can use this information to decide whether more data generation (by performing experiments) or learning is required. Thus, QPOLE proves to be an effective surprise detection mechanism for open-ended learning in an embodied discoverer.

The parameters of the mechanisms used by the temporal abstraction provided satisfactory results for the experiments presented here. It is obvious though that the same parameters are not valid for every case. An erroneous parameter selection may result in an incorrect signal approximation to the sensor input, an excessive delay in recognizing changes in the trends, or even the complete absence of such changes. We believe that a self-adaptive mechanism which is able to learn from past experiences, as well as the context of the current perception, will help in determining the correct parameters for observing the outcome of different kinds of actions.

Our future work on QPOLE will focus on developing this self-adaptivity within the surprise mechanism, and on introducing semi-quantitative differential equations. Futhermore, we plan to extend the Observation mechanism so that multiple concurrent hypotheses, represented by different models and behavior trees, can be tested simultaneously. Other issues to be adressed are periodic or semi-periodic behavior of the observed objects, which become an important aspect in more complex experiments.

REFERENCES

[1] E. Prassler, A. Juarez, and T. Henne, "Robotic discovery and learning by experimentation: an overview," Univ. of Applied Sciences Bonn-Rhein-Sieg, Tech. Rep., 2007.

[2] J. Schmidhuber, "Curious model-building control systems," in *International Joint Conference on Neural Networks*, vol. 2, 1991, pp. 1458–1463.

[3] M. Peters, "Towards artificial forms of intelligence, creativity, and surprise." in *Proceedings of the 20th. Meeting of the Cognitive Science Society*, 1998, pp. 836–841. [Online]. Available: citeseer.ist.psu.edu/peters98towards.html

[4] L. Macedo and A. Cardoso, "Towards artificial forms of surprise and curiosity," in *Proceedings of the European Conference on Cognitive Science*, S. Bagnara, Ed., 1999, pp. 139–144. [Online]. Available: citeseer.ist.psu.edu/509729.html

[5] H. Kay, B. Rinner, and B. Kuipers, "Semi-quantitative system identification, Tech. Rep. AI99-279, 8 1999. [Online]. Available: http://www.cs.utexas.edu/users/qr/papers-QR.html

[6] R. G. Sargent, "Verification and validation of simulation models," in *Winter Simulation Conference*, 1998, pp. 121–130.

[7] L. Ljung, *System identification: theory for the user*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986.

[8] ——, "Identification for control – what is there to learn," in *Proc. Workshop on Learning, Control and Hybrid Systems Bangalore, India, January 1998.*, 1998.

[9] U. Forssell, "Properties and usage of closed-loop identification methods," Licentiate thesis, Sep 1997, LIUTEK-LIC-1997:42, Department of Electrical Engineering, Linkoping University, Sweden. [Online]. Available: citeseer.ist.psu.edu/forssell97properties.html

[10] B. J. Kuipers, "Reasoning with qualitative models." *Artificial Intelligence*, vol. 59, pp. 125–132, 1993.

[11] B. Kuipers, "Qualitative simulation," *Artificial Intelligence*, vol. 29, pp. 289–338, 1986. [Online]. Available: http://citeseer.ist.psu.edu/kuipers86qualitative.html

[12] J. J. Flores and N. Pastor, "Qualitative systems identification for linear time invariant dynamic systems," in *Mexican International Conference on Artificial Intelligence MICAI*, 2002, pp. 470–476. [Online]. Available: citeseer.ist.psu.edu/article/flores02qualitative.html

[13] P. Salles, B. Bredeweg, S. Araujo, and W. Neto, "Qualitative models of interactions between two populations," *AI Communications*, vol. 16, no. 4, pp. 291–308., 2003. [Online]. Available: citeseer.ist.psu.edu/article/salles03qualitative.html

[14] D. Dvorak and B. Kuipers, "Model-based monitoring of dynamic systems," in *Proc. 11th Int. Joint Conf. on Artificial Intelligence (IJCAI-89)*. San Mateo, CA: Morgan Kaufmann, 1989, pp. 1238–1243. [Online]. Available: citeseer.ist.psu.edu/dvorak89modelbased.html

[15] B. Kuipers and D. Berleant, "Using incomplete quantitative knowledge in qualitative reasoning," *National Conference on Artificial Intelligence*, 1988.

[16] D. Berleant and B. Kuipers, "Qualitative-numeric simulation with q3," *In Boi Faltings and Peter Struss (Eds.), Recent Advances in Qualitative Physics, MIT Press*, 1992.

[17] T. min Lee, U. Nehmzow, and R. Hubbold, "Computer simulation of learning experiments with autonomous mobile robots," in *Proceedings of Towards Intelligent Mobile Robotics (TIMR), Bristol, 1999*. [Online]. Available: citeseer.ist.psu.edu/224323.html

[18] B. Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* Cambridge, MA: . MIT Press, 1994.

[19] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, no. Q2, p. 15, 1998. [Online]. Available: citeseer.ist.psu.edu/bradski98computer.html

[20] S. Miksch, W. Horn, C. Popow, and F. Paky, "Context-sensitive and expectation guided temporal abstraction of high-frequency data," Austrian Research Institute for Artificial Intelligence, Vienna, Tech. Rep. TR-96-02, 1996. [Online]. Available: citeseer.ist.psu.edu/miksch96contextsensitive.html

[21] R. Bellazzi, P. Magni, C. Larizza, G. D. Nicolao, A. Riva, and M. Stefanelli, "Mining biomedical time series by combining structural analysis and temporal abstractions," *Journal of the American Medical Informatics Association (JAMIA), Symposium supplement*, vol. 5, pp. 160–164, 1998. [Online]. Available: citeseer.ist.psu.edu/bellazzi98mining.html