Q-POLE: A QUALITATIVE PREDICTION-OBSERVATION LOOP FOR LEARNING BY EXPERIMENTATION

Ashok Mohan, Alex Juarez, Timo Henne

Project XPERO: Learning by Experimentation
University of Applied Sciences Bonn-Rhein-Sieg
Grantham Allee 20, D-53757, Sankt Augustin, Germany

xpero-brs@xpero.org(XPERO)

Abstract

An agent that learns by experimentation 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 *Q-POLE* which stands for Qualitative Prediction-Observation loop for Learning by Experimentation. The *Prediction Engine* of this loop uses a qualitative model of the world to predict temporal states of the system. For this, attributes and their domain, start state, elementary behaviors and the stop criteria of the model are defined by using Qualitative Differential Equations (QDEs). These QDEs are then used to generate a qualitative state tree called *behavior tree* which consist of the possible temporal states that the system can attain. The Prediction Engine uses these behavior trees for generating the qualitative prediction, while the *Observation Engine* 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 reflects the results of the agents actions.
In this paper we present and discuss the results of first experiments with Q-POLE that established and tested the thresholds for some simple real-world setups with a rolling and bouncing ball.

Keywords: Learning by experimentation, prediction, observation, qualitative simulation, Qualitative Differential Equations, behavior trees, temporal abstraction, surprise.

Presenting Author's Biography

Timo Henne was born in 1974 in Kiel, Germany. He received his diploma in computer science from the University of Bonn in July 2005. In his diploma thesis supervised by the Division of Neuroinformatics he developed an adaptive action selection mechanism based on an internal value system for an autonomous mobile robot. He is currently employed by the University of Applied Sciences Bonn-Rhein-Sieg. The current focus of his work within the EU project XPERO (FP6-IST-29427) includes internal value systems, hypothesis revision, knowledge representation and design of experiments.

ISBN 978-3-901608-32-2

1 Introduction

The work presented in this paper was carried out in the context of XPERO (FP6-IST-29427), a project that aims to enable an embodied agent to augment its cognitive capabilities through open-ended learning by experimentation. This type of learning is a form of "Learning through interaction with the world", which we view as complementary to the concepts of "Learning through introspection" and "Learning through interaction with an instructor". The goal of XPERO is to enable the agent to autonomously design and conduct experiments in order to achieve, maintain and improve a consistent model of its environment and of the objects and its own embodiment therein.[1]

This objective demands for a possibility for the agent to verify the hypotheses which form its world model in a non-experimental context. If a significant deviation between the predicted and the actual result is encountered, the agent will have to revise its hypotheses by designing, planning and executing experiments. In the beginning, the agent's hypotheses will not be precise formulas modelling physical phenomena, since these are often far too complex for the robot to derive from its sensor data. Instead, the environment will be represented by qualitative models, which offer the advantage of providing more expressive power of incomplete knowledge than most numerical methods. The remaining challenge is to effectively compare qualitative process models to numerical data acquired during observation.

The prediction and observation of processes has been thoroughly studied by disciplines such as system dynamics, control theory and statistics. In this context, system identification is the process of combining a partially-specified model with observations from a system to converge on a more accurate and precise model. Closed-loop variants are available such that the system can be modeled and controlled even with only a sparse knowledge of it. At the same time, model validation is necessary to assure that the model can successfully explain a situation in the real world and provide the appropriate feedback to close the loop.[2] Qualitative approaches in this area focus mainly on extracting qualitative models from observed data.[3, 4, 5, 6]

Computer simulation also has explored the subject in different domains. Lee et al. use a model of the sensory perception of a learning agent obtained from the interaction of the agent and the environment. The results of this model are 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.[7]

SQUID is a system identification method which attempts to match semi-quantitative trends to semi-quantitative behaviors [8]. For this, the space of potential models is defined by semi-quantitative differential equations. SQUID uses a conservative refinement technique which eliminates only those ODEs from the model space which do not fit the observation data. While this approach seems applicable for certain domains, it is higly limited when applied to measurements that are a superposition of Gaussian noise with fixed variance to the pure signal. Furthermore, it operates as a batch computation over the measurement stream and not as an online prediction-observation system.

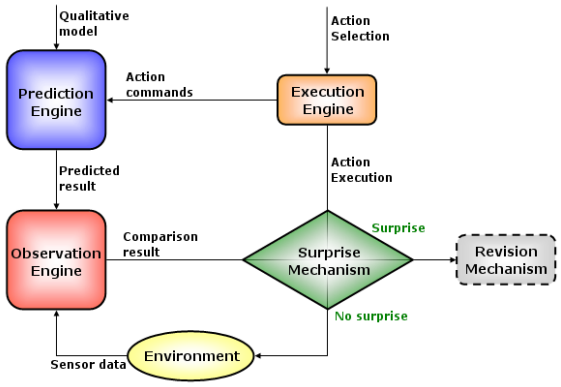

Fig. 1 The Prediction-Observation loop Q-POLE. A qualitative model of the experimental setup is used to predict the behavior of the system. The sensor-obtained observation from a real environment is then compared to this prediction eventually yielding a surprise.

Our proposed prediction-observation loop Q-POLE establishes a possiblity for online comparison of qualitative models and hypotheses to numerical sensor data gathered by an autonomous robot in a real world environment. For this purpose the data acquired during observation are abstracted to qualitative values by an *Observation Engine*. The comparison to the qualitative prediction from the *Prediction Engine* then yields a measure of mismatch which is passed on to an adaptable mechanism. In the case of a significant deviation from its current model, this *Surprise Mechanism* blocks the scheduled execution of actions and triggers an external mechanism either for revising the prediction model, or for repeating the last action or action sequence to avoid erroneous results caused by noise. In the case of a sufficiently accurate prediction, the model is confirmed and the execution of actions continues. Figure 1 depicts the overall concept of Q-POLE.

In sections 2 and 3 of this paper, we describe the techniques and mechanism used in Prediction and Observation Engines respectively. Section 4 illustrates our first experiments with Q-POLE for testing its effectiveness, which is discussed in section 5.

2 Prediction

Prediction attempts to claim that a certain event will occur in the (near) future. In order to achieve this, the prediction engine uses the simulation software QSIM [9] to perform qualitative simulation of the manually written *qualitative model*. QSIM generates a behavior tree of states starting from a pre-specified initial state. This behavior tree is used by the prediction engine to define the future events. It is also used to create an appropriate data structure that facilitates comparison with

ISBN 978-3-901608-32-2

the observations.

2.1 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 are similar to ODEs but are able to express more incomplete knowledge. Qualitative Differential Equations (QDEs) provide an efficient way to define a qualitative model. A QDE model can be viewed as another layer of abstraction over 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. [9, 10]

2.2 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.

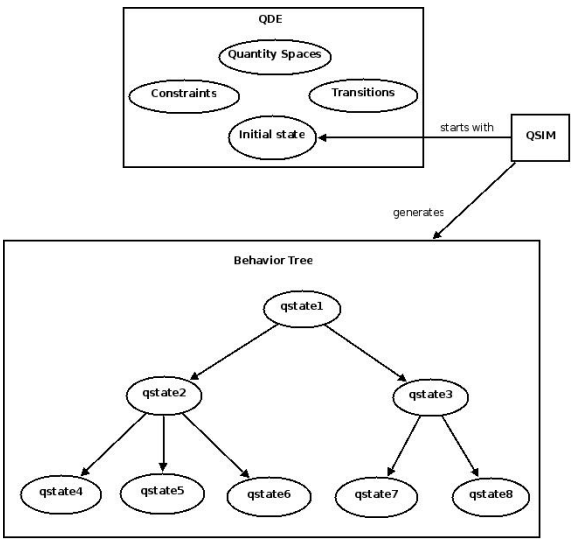

Fig. 2 The prediction process

The Figure 2 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 is done until all the variables reach a steady state or until a pre defined number of levels. Every path starting from the root of the behavior tree until a leaf node is a possible qualitative behavior.[9, 10]

The behavior tree is a tree of possible temporal states achievable by the defined model. The links of the tree are transitions from one state to another and the nodes are the temporal states of the model. The nodes contain the qualitative values of each variable defined in the model. An appropriate transformation is required to ensure easy comparison with the observations recorded. In order to ensure this, the values are extracted by the prediction engine and pre processed. The tree of behaviors are sent to the observation engine which compares these temporal states to the real observed data.

3 Observation

The perception of the environment in which an agent is situated depends on the sensors it is equipped with. Visual sensors provide a wide range of possibilities to analyze and transform incoming data into meaningful information useful to evaluate the outcome of an experiment. However, due to the nature of such sensor, the information obtained is always quantitative. This quantitative information is abstracted to qualitative values to facilitate comparison with the prediction.

The environment is monitored using color and motion tracking. Color histograms are obtained from the objects that will be tracked and the CAMSHIFT algorithm [11] is applied to obtain the position, velocity acceleration and approximate size of the object at a specific time.

3.1 Surprise

An observation that concurs with the prediction is interpreted as a confirmation of the current model. A mismatch between prediction and observation refutes the current model thereby triggering surprise. The surprise and the circumstances under which it occured (e.g. action, time, feature preconditions etc.) can then be used by another mechanism for eventually revising or changing the prediction model. The loop continues as long as no surprise is detected (see fig. 1).

3.2 Temporal Abstraction

The extraction of values from the observed quantitative data that are comparable to the qualitative values provided by the prediction model is central to our approach. The substantial amount of data obtained over time from monitoring the robot actions needs to be abstracted to a new representation meaningful for comparison. Such representation may include trend templates, temporal interpolation, temporal inference among others [12]. The concept described is also known as temporal abstraction and has been successfully applied to varied domains such as medical information analysis,

ISBN 978-3-901608-32-2

hierarchical and abstract planning and reinforcement learning. Most techniques for temporal abstraction and comparison of qualitative and/or quantitative values are known to work over offline data sets, e.g. monitoring of blood glucose in diabetic patients from controls of past months [13].

An interesting approach to qualitative-quantitative comparison is presented by Kay et al. [8], where a temporal abstraction is performed by a "sliding window" that ran through the data points over time obtaining the slope of the best fitting line, finding the trends of the data distribution. Though in principle, the technique could be applied to online data, the results presented denoted that the data used had been obtained a priori.

We present a qualitative-quantitative comparison approach that builds upon the concepts and ideas presented before. We propose a method that extracts trends from online data coming from the robot sensors, and compares them with the predicted trends produced by the prediction engine. The trend analysis is bivariate with one of the variables used as a common base, usually time, and the other as the compared variable.

Our method assumes the input received from sensors as a signal that must be processed. For this matter we incorporate techniques from time series analysis such as a weighted moving average that allows a reconstruction of the signal measured. Such reconstruction offers the advantage of a better approximation of the original signal with greater flexibility at defining the window size. This is depicted by Formula 1

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

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.

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 Formula 2

$$y(t) = x(at - b) \qquad (2)$$

where $a \in \Re$ and $b \in \aleph^+$. 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.

3.3 Trend matching

The temporal abstraction that is applied to the incoming data from sensors, provides the current system behavior trend resulting from the robot actions. A comparison between the current predicted and observed trend is performed. If a mismatch is detected, the next predicted trend is examined. If there is a match between trends, the observation continues, otherwise surprise is triggered and the observation ends. In the case when there is no surprise, there are two stop criteria:

* all qualitative predicted behaviors have been observed or,
* the observation time has expired.

Such observation time is fixed prior to the experiment

4 Experiments

The experimental setup is composed of a rectangular area of 1.20 x 0.80 meters enclosed by walls 20 cm in height. A color CCD camera with a resolution of 640x480 pixels and a framerate of 30 fps is mounted on top of the robot pointing in the direction of the robot's orientation. The floor of the environmental setup is of grey color while the walls are white. The ball used to perform the experiments is colored red. The robot is equipped with a gripper that enables it to lift the ball from the ground. Thus the robot can actively perform the experiments: rolling (by pushing it) and vertical bouncing (by lifting and dropping it).

The observation time is not fixed, however, at the current state of our work the human expert decides when to stop observing in case there is no surprise. This can be enhanced to use other mechanisms like visual attention, boredom and curiosity to start and stop an observation cycle. The parameters for the temporal abstraction were obtained empirically from observations. The weighted moving average considers a window of five observations with a decreasing weight that emphasizes older inputs. The hysteresis delays the detection of change in the trend by 15 observations (approx. half a second).

Multiple experimental scenarios were used to to test the performance of Q-POLE. In the first scenario, the robot pushes the ball which comes to a stop after rolling for some time, owing to friction and air resistance. The second scenario is similar to the first one, but here an obstacle 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 scenario, the robot lifts the ball and drops it, thus making it bounce vertically.

4.1 Scenario 1: Rolling ball.

In this scenario 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 size of the ball. From both, the variable with respect to time is the size of the ball.

The prediction generates the behavior tree shown in figure 3, where the variable values in parenthesis stand for the following:

- *smax*: qualitative indicator of the actual size of the ball
- *smin*: any perceived size between zero and smax
- *inf*: infinitive value
- *inc /dec /std*: increasing/decreasing/steady value

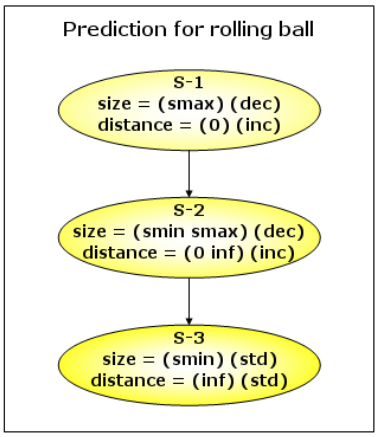

Fig. 3 The behavior tree generated by the prediction engine for a ball rolling and stopping due to friction, with the variables `size` of the ball and `distance` between robot and ball. The terms in brackets behind each variable specify the qualitative value and the qualitative trend in each state.

In the case of two values in one parenthesis (like in state S-2), this indicates that the variable in this state will have a value in between the two given qualitative values.

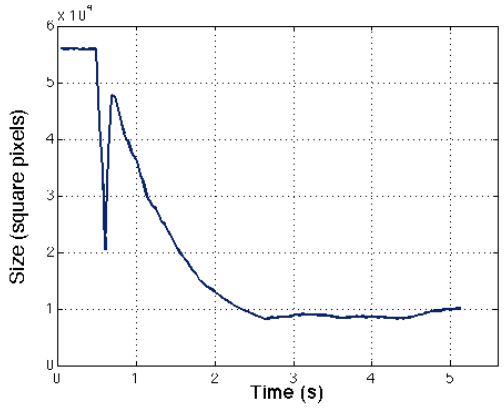

Fig. 4 Observed area (in squared pixels) of the color blob that represents the ball that the robot sees. The observation starts the instant before the robot pushes the ball (approx. 1 second) and finishes when the ball has stopped (approximately 5 seconds).

In this scenario the prediction generates 3 consecutive states: S1-S2-S3. According to the predicted behavior, the ball rolls for a while and then comes to a halt because of air resistance and friction. 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 detects the color blob associated with the ball (red) and calculates its area, measured in pixels.

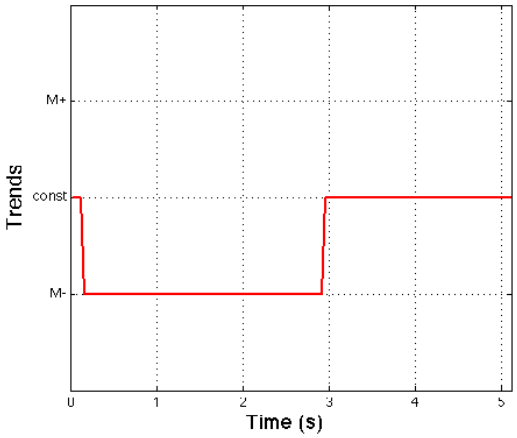

Fig. 5 Visualization of the trends extracted using the time abstraction mechanism for this particular scenario. The trends are consistent with the numerical observations: an initial constant value is observed in the instant before the robot pushes the ball, followed by a monotonically decreasing trend of the area value as the ball rolls away.

The time basis for the observation is the video framerate (approx. 25 fps after image processing). Figure 4 shows the variation in object size with respect to time taken across approx. 5 seconds of observation of the event.

The time abstraction mechanism allows to reduce the effect of noise in the observation. In this case, trend matching does not detect any deviation in the signal from the qualitative prediction and thus, surprise is not triggered and the robot continues with its normal tasks. Figure 5 presents a visualization of the trends extracted from the data mentioned above and used in the matching.

4.2 Scenario 2: Rolling ball bouncing off the wall.

This scenario was intended to test the model that was created for the previous one. The observation was made under the same conditions. The robot pushes the ball as in the previous scenario, but an obstacle is introduced manually such that the ball rebounds off it and rolls back towards the robot. The observed behavior of the ball size variable is shown in Figure 6. 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, Q-POLE triggers a surprise. This may 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. Within XPERO, the result of these external mechanisms will be used to refine the model so that future predictions may concur with it.

A visualization of the temporal abstraction for the case where there is no surprise is shown in Figure 7. In our experiment, the model is not able to correctly predict the behavior of the observed variable once this starts

ISBN 978-3-901608-32-2

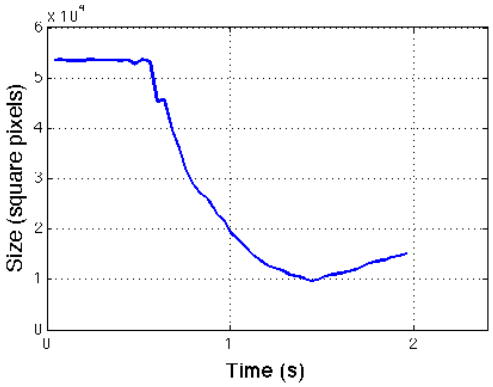

Fig. 6 Observed area (in squared pixels) of the color blob that represents the ball that the robot sees. The observation starts the instant before the robot pushes the ball. The ball rolls until it bounces off the wall. The observation stops when the perceived data does not match the predicted behavior.

increasing as a result of the bounce. So a surprise is triggered by the surprise mechanism.

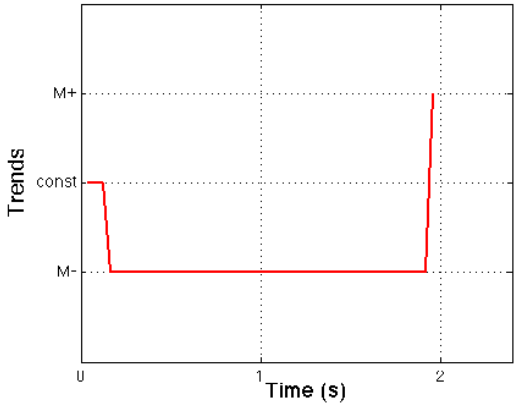

Fig. 7 Visualization of the trends extracted using the time abstraction mechanism for this particular scenario. In our experiment, we used the model built for Scenario 1, resulting in surprise triggered (trend change to increase).

4.3 Scenario 3: Drop ball.

Scenario 3 demonstrates how surprise is obtained as a result of a model that is correct for the ideal case, but falls short when applied to real observations. Following the procedure described previously, we created a qualitative model for the drop ball scenario. The variables that are observable over time are the 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 (measured in pixels per time step) of the ball as it bounces.

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

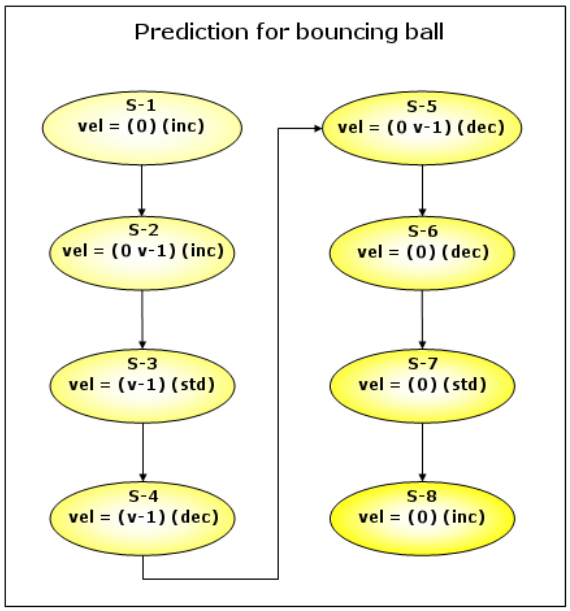

Fig. 8 The behavior tree generated by the prediction engine for a ball bouncing, with `vel` as the variable for velocity.

learn by refining the model, using semi-quantitative description which is not yet realized within Q-POLE. The prediction generates a behavior tree as shown in Figure 8. As can be seen in the figure, the prediction generates a behavior that predicts the ball bouncing vertically at the same location with change in the direction of velocity. This behavior is the one that is compared online by Q-POLE with the observation made in this experiment.

For this model, both variables trigger surprise as they deviate from the predicted behavior (Figure 9). 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. Such prediction states that a steady (constant) velocity is reached in each contact with the floor and return to original drop position. The observation of such a state proved to be very difficult, especially in cases when 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 accounted for, temporarily skipped when looking for the next node in the behavior tree. Figure 10 shows the trends obtained before surprise was triggered.

It is clear from the results that our model wasn't accurate enough to account for the situations that a robot

ISBN 978-3-901608-32-2

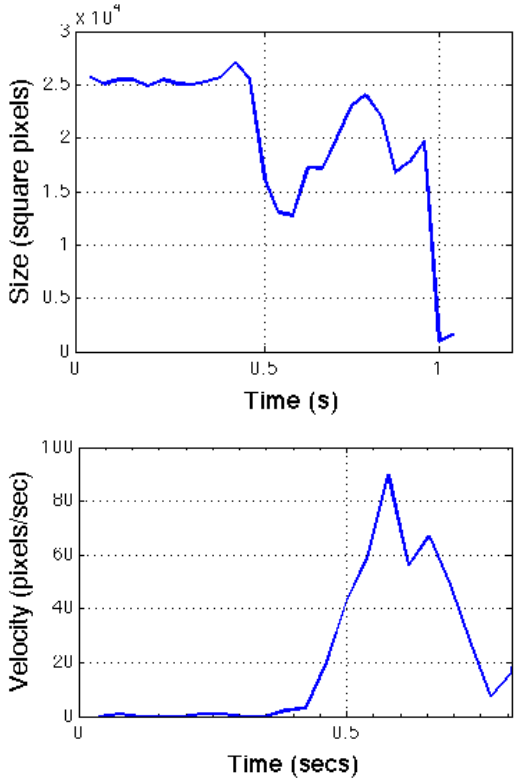

Fig. 9 Behavior of ball size and ball velocity over time for the drop ball scenario. The size of the ball is measured in square pixels . The velocity is measured in pixels per milliseconds. The scale of the figures is shown in seconds.

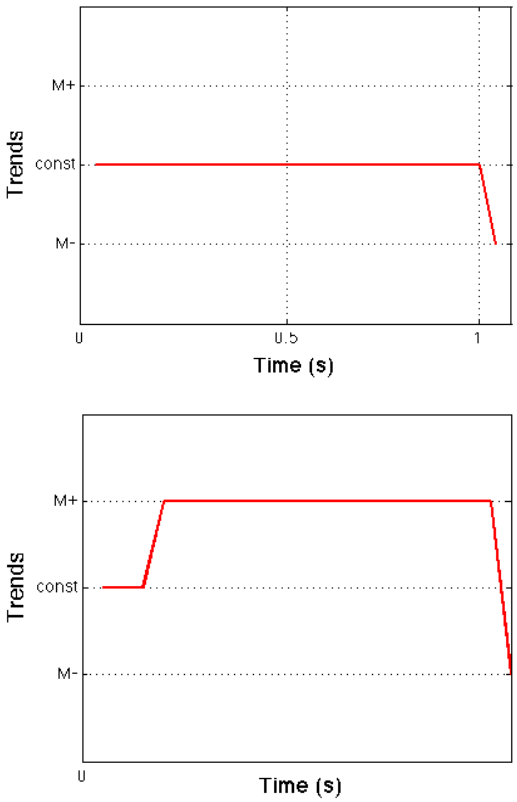

Fig. 10 Visualization of the trends extracted using the time abstraction mechanism for this particular scenario. In our experiment, the model used predicted a constant value for the variable size as well as a continuous sequence of increasing, steady and decreasing values for velocity. This resulted in a surprise triggered by the decreasing and oscillatory trends observed respectively.

has to deal in a real environment. These are the characters of the system that the robot has to gradually learn. The data from the observation stream that triggered this surprise maybe used to revise the model appropriately. An improvement would be to compare the observations with behaviors from multiple models and use exclusion to narrow the possible models of the system.

5 Discussion

The current qualitative representation 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 or include phenomena such as friction in more complex scenarios such as the drop ball scneario.

The surprises triggered by Q-POLE can be passed on to external mechanisms which can use this information to decide whether more data generation (by performing epxeriments) or learning is required. Thus, Q-POLE 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 scenarios 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 that 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 Q-POLE 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 Engine 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 will become an important aspect in more complex scenarios.

ISBN 978-3-901608-32-2

6 Acknowledgement

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").

7 References

[1] Erwin Prassler. Learning by experimentation: Xpero project proposal, 2005.

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

[3] Juan J. Flores and Nelio Pastor. Qualitative systems identification for linear time invariant dynamic systems. In *MICAI*, pages 470–476, 2002.

[4] P. Salles, B. Bredeweg, S. Araujo, and W. Neto. Qualitative models of interactions between two populations, 2003.

[5] Mor Peleg, Irene S. Gabashvili, and Russ B. Altman. Qualitative models of molecular function: Linking genetic polymorphisms of tRNA to their functional sequelae. Proceedings of the IEEE, 2002.

[6] S. M. Garrett, G. M. Coghill, A. Srinivasan, and R. D. King. *Learning Qualitative Models of Physical and Biological Systems*. 2004.

[7] Ten min Lee, Ulrich Nehmzow, and Roger Hubbold. Computer simulation of learning experiments with autonomous mobile robots.

[8] Herbert Kay, Bernhard Rinner, and Benjamin Kuipers. Semi-quantitative system identification. Technical Report AI99-279, 8 1999.

[9] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

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

[11] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2):15, 1998.

[12] S. Miksch, W. Horn, C. Popow, and F. Paky. Context-sensitive and expectationguided temporal abstraction of high-frequency data, 1996.

[13] R. Bellazzi, P. Magni, C. Larizza, G. De Nicolao, A. Riva, and M. Stefanelli. Mining biomedical time series by combining structural analysis and temporal abstractions, 1998.

