I. STATEMENT OF COMMITMENT

Team Noxious-Kouretes is committed to participate to RoboCup 2011. Funds have been secured for participation.

II. TEAM HISTORY

Team Noxious-Kouretes is a joint team formed by three universities (Newport, Oxford, TUC) in two countries (UK, Greece). This collaboration emerged naturally following a recent RoboCup event (RoboCup Exhibition and Engagement Event, Eisteddfod of Wales 2010) where all three met and a recent move of personnel between groups. In addition, the fact that there is almost no overlap between the research work of the three groups further motivated the joining of forces towards a strong and competitive joint team. Even though this is the first time the three groups apply as a joint team, each member group has a history of RoboCup participation.

Team Noxious is a collaboration between the Cognitive Robotics Research Center (CRRC), University of Wales, Newport and the Computing Laboratory (ComLab), University of Oxford. The team was formed only in May 2010, but soon after, in August 2010, the team organized and hosted UK’s first official RoboCup event RC4EW, featuring Standard Platform and 3D Simulation leagues. Two members of Noxious team are part of the newly created NAOSShare community which consists of 30 European professors and students sharing academic resources engaging the Nao robot in educational programs. The team has received several distinctions in the 2D/3D Simulation League under the name OxBlue: runner-up in UK FIRA competition 2007; 8th place in 2D Sim, 12th place in 3D Sim at RoboCup 2008; 2nd place in 2D Sim at RoboCup 2009; and 3rd place in 3D Sim at RC4EW 2010.

Team Kouretes was founded in 2006 and participates in the main RoboCup competition ever since in various leagues (Four-Legged, Standard Platform, MSRS, Webots), as well as in various local RoboCup events (German Open, Mediterranean Open, RC4EW, RomeCup) and RoboCup exhibitions (Athens Digital Week, Micropolis, Schoolfest). In May 2010, the team hosted the 1st official SPL tournament in Greece (with three invited teams) within the Hellenic Conference on Artificial Intelligence (SETN). The team has been developing its own (publicly-available) software for the Nao robots since 2008. Distinctions of the team include: 2nd place in MSRS at RoboCup 2007; 3rd place in SPL-Nao, 1st place in SPL-MSRS, among the top 8 teams in SPL-Webots at RoboCup 2008; 1st place in RomeCup 2009; 6th place in SPL-Webots at RoboCup 2009; and 2nd place in SPL at RC4EW 2010.

III. TEAM LEADERSHIP AND MEMBERS

Torbjørn S. Dahl is a Reader in Cognitive Robotics at the University of Wales, Newport and Founder and Leader of the Cognitive Robotics Research Center. He was awarded an MEng from Imperial College, London in 1997 and a PhD from the University of Bristol in 2002. Dahl was a post-doc at the University of Southern California in 2001/02 and has also worked for Hewlett-Packard Labs, Bristol, and the Norwegian Defense Research Establishment. He is a member of IEEE, ACM, and an executive committee member for the IET TNP Robotics and Mechatronics. Dahl has been a PI on both EPSRC and FP7 projects and he is the lead academic on a collaborative POWIS project with Sony. Dahl was the organizing committee chair for the RC4EW 2010 event.

Stephen Cameron is a Reader in Computing Science at Oxford University. He obtained his PhD in Artificial Intelligence at Edinburgh University, working on the geometric modeling of robots and on collision detection. His general area of interest is in spatial reasoning, covering a wide range which includes the planning of tasks and motions for robot vehicles and manipulators, the use of geometric models, and the scheduling of fleets of robots. He is a member of the AISB, the ACM, the IEEE RAS, the Geometric Modelling Society, the IAM and CAMRA.

Michail G. Lagoudakis is an assistant professor at TUC since 2005. He received his Ph.D. from Duke University, USA in 2003 and was a postdoc at Georgia Tech, USA until 2005. His research experience in robotics includes path planning, motion control, reinforcement learning, coordination. He is a member of the SPL Executive Committee.
since 2009 and was co-chair of the 13th RoboCup Symposium. **Nikos Vlassis** is an assistant professor at TUC since 2007. He received his Ph.D. from the Technical University of Athens, Greece in 1998 and was an assistant professor with the University of Amsterdam, Netherlands until 2006. His research interests include stochastic optimal control, unsupervised learning, and reinforcement learning. Vlassis has various distinctions in the Simulation league with the UvA Trilearn team: 1st at RoboCup 2003, 1st at German Open (2003, 2004, 2005), and 1st at US Open (2003).

The joint team Noxious-Kouretes 2011 counts **12 members** (brackets indicate the main area of each member):

<table>
<thead>
<tr>
<th>Name</th>
<th>Role/Position</th>
<th>Institution</th>
<th>Main Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eleftherios Chatzilaris</td>
<td>graduate student</td>
<td>TUC</td>
<td>Localization</td>
</tr>
<tr>
<td>Iris Kyranou</td>
<td>undergraduate student</td>
<td>TUC</td>
<td>Obstacle Avoidance</td>
</tr>
<tr>
<td>Jie Ma</td>
<td>postdoc researcher</td>
<td>Oxford, ComLab</td>
<td>Walk Learning</td>
</tr>
<tr>
<td>Emmanuel Orfanoudakis</td>
<td>undergraduate student</td>
<td>TUC</td>
<td>Visual Object Recognition</td>
</tr>
<tr>
<td>Andreas Panakos</td>
<td>graduate student</td>
<td>Newport, Sony</td>
<td>Color Recognition</td>
</tr>
<tr>
<td>Alexandros Paraschos</td>
<td>laboratory staff</td>
<td>Newport, CRRC</td>
<td>Software Architecture</td>
</tr>
<tr>
<td>Georgios Perris</td>
<td>graduate student</td>
<td>Newport, CRRC</td>
<td>Motion Skill Learning</td>
</tr>
<tr>
<td>Nikolaos Spanoudakis</td>
<td>laboratory staff</td>
<td>TUC</td>
<td>ASEME Methodology</td>
</tr>
<tr>
<td>John Threlfall</td>
<td>undergraduate student</td>
<td>Newport, CRRC</td>
<td>Object Recognition</td>
</tr>
<tr>
<td>Aggeliki Topalidou-Kyniazopoulou</td>
<td>undergraduate student</td>
<td>TUC</td>
<td>Behavior, Coordination</td>
</tr>
<tr>
<td>Dimitra-Astero Tzanetatou</td>
<td>undergraduate student</td>
<td>TUC</td>
<td>Behavior, Coordination</td>
</tr>
<tr>
<td>Evangelos Vazaios</td>
<td>graduate student</td>
<td>TUC</td>
<td>Robot Communication</td>
</tr>
</tbody>
</table>

**IV. TEAM RESEARCH AND PLANNED ACTIVITIES**

This section presents briefly some of the team’s recent research, focusing only on the interests of the current members. Where appropriate, a short description of planned activities is provided. For a comprehensive account of the joint team’s research, please check the reports and papers on each group’s web site.

**A. Software Architecture and Communication (TUC/Newport)**

The team’s code is based on MONAS, a software architecture developed in-house to address the needs of the team. MONAS provides an abstraction layer from the Nao robot and allows the synthesis of complex robotic teams in various ways: (a) NaoQi modules, (b) XML-specified MONAS agents, and (c) model-based (ASEME) statechart agents. Apart from the trivial method (a), method (b) allows the coding of different agents (e.g. perception, motion, behavior, etc.) each executed independently at any desired frequency completing a series of activities at each execution. Method (c) is based on the Agent Systems Engineering Methodology (ASEME), whereby the target team behavior is specified through a series of model-based transformations as a statechart, which is executed using a generic multi-threaded statechart engine. A high-level view of MONAS is shown in Figure 1 (left).

To realize a fully-functional robot team, MONAS relies on our intra-robot and inter-robot messaging system, called NARUKOM. NARUKOM is based on the publish/subscribe paradigm and provides maximal decoupling not only between nodes, but also between threads on the same node. The data shared between nodes/threads are stored on local blackboards which are transparently accessible from all nodes/threads. Communication is achieved through messages tagged with appropriate topics and relayed through a message queue which is implemented using Google protocol buffers. Three types of messages are supported: (i) *state messages*, which remain in the blackboard until they are replaced by a newer message of the same type, (ii) *signal messages*, which are consumed at the first read, and (iii) *data messages*, which are time-stamped to indicate the precise time the values they carry were acquired. Data messages have expiration times, so that they are automatically removed when they are no longer needed. NARUKOM’s distributed nature and platform independence make it an ideal base for coordinating asynchronous modules, for developing complex team strategies, and for relaying debugging information to external computers.

**B. Vision and Localization (TUC/Newport)**

The objects of interest in the SPL are characterized by unique colors. Our approach to color recognition is based on labeling by hand a representative set of images from the robot camera, training a classifier which generalizes over the entire color space, and generating a static color map for constant-time recognition during deployment. This process is supported by our KC² graphical tool. Since light sources affect significantly the correct recognition of
colors, the white and gray surfaces of the Nao body are utilized for providing a reference point for the illumination of the environment and for calibrating the camera’s white balance dynamically. This dynamic camera calibration enabled us to use a single color recognizer trained at RoboCup 2010 in a wide range of lighting environments.

Object recognition is accomplished by KV\textsc{vision}, a light-weight image processing scheme which applies color recognition to a selected set of pixels directly on the YUV images provided by the hardware to locate areas of interesting colors. These areas are then processed locally and examined further for validity as field objects. The exact camera position in the 3-D space (extracted from the robot’s kinematic chain) is utilized to determine the sampling grid, so that scanning is uniformly projected over the field, not over the image matrix. Special attention is paid in synchronizing precisely images with robot joint values using time-stamped data messages. Kinematics and camera position are also used to determine the view horizon, the expected projected size of the known-size SPL objects on the image, and the head joint values for fixating on objects of interest. Multiple matches for various objects (e.g. multiple balls) are evaluated and filtered, so that the best match (if any) in terms of color, shape, and size for each unique object is finally extracted and returned as perceived, along with an estimate of its distance and bearing as well as head joint information for fixation and tracking. Figure 1 (right) shows an example with the raw and the segmented image along with the perceived objects.

Self-localization of the robots in the field is accomplished using KL\textsc{oc} which realizes Monte Carlo localization with particle filters. The belief of the robot is a probability distribution over the 3-dimensional space of \((x, y, \theta)\), where \(x, y\) are the robot coordinates on the field from a bird’s eye view and \(\theta\) is the orientation of the robot with respect to the line that crosses the center of the goals. The belief is represented approximately using a population of particles. In order to perform belief update, we use an odometry motion model that covers all possible omnidirectional locomotion choices and a landmark sensor model for the yellow/blue goalposts over the \((x, y, \theta)\) space. The parameters of both models were estimated through extensive experimentation. Belief update is performed using an auxiliary (AUX) particle filter. For the resampling process, selection with replacement and linear-time resampling have been implemented. Given any population of particles, the robot’s pose is estimated as the pose of the particle with the highest weight. Figure 1 (right) shows our localization interface which displays the perceived landmarks (two yellow goalposts) and the estimated robot position.

C. Walk Learning (Oxford)

Our Q\textsc{walking} method is a careful application of the popular reinforcement learning technique \(Q\)-learning to the problem of learning robot walking patterns. The advantage of using a learning technique over analytical methods is twofold: we can deal with imprecise models of the robot’s dynamics or kinematics; and we can adjust the learned actions as the robot wears. The major difficulty in applying \(Q\)-learning to this task is the potential size of the state space and thus the slow speed of learning convergence. We deal with this by using a simplified gait model with a small number of parameters and by first applying the learning within a physically accurate simulation of the robot.
(the SPARK simulator). The former gives us a state space of manageable size (a few thousand states), while the latter greatly decreases the overall learning time, not least because the robot is usually dynamically stable when the learning phase switches over to use on the real robot. Finally, by applying an incremental $Q$-learning refinement technique, we can in principle refine the walking patterns as the robot ages. QWALKING has already been used for producing walking patterns for use in the 3D simulation league. Figure 2 shows a description of the leg swing phase (left) and an example of a continuous gait pattern generated by QWALKING (right). For use in SPL, we transferred the refined fast walking patterns from the 3D Simulation directly onto the real Nao robots. The speed of the fast QWALKING is approximately 22 cm/sec, which roughly doubles the original Aldebaran walking speed (11.3 cm/sec). Our future plan include omnidirectional walking and balancing walking speed and energy-efficiency.

**D. Motion Skill Learning (Newport)**

Motion skills, such as player kicking or goalkeeper diving, are crucial for a RoboCup team. The majority of teams use manually-designed, pre-recorded motions, which are fine-tuned on spot for optimized performance. Despite the drawback of these approaches, automatic skill acquisition/learning is not widely used, due to long learning periods and robot durability/safety issues. Our team currently makes a transition from manual motion design (e.g. our past work on the KME tool) to learned motions focusing on biologically-inspired approaches to learning gestures and motion skills on humanoid robots. According to neuropsychologists, infant cognition can be modeled as a hierarchical bottom-up structure. Our algorithms are based on hierarchical temporal structuring of states and actions using fixed-length data sequences on each level of the hierarchy. Hierarchy is achieved using hierarchical self-organizing maps (SOMs), and temporally is achieved using decaying activity levels for winning SOM nodes. This structure massively compresses temporal data through the reuse of common sub-sequences in the hierarchy and through a principle component-based representation of each fixed sequence. We have successfully used this structure to learn and reproduce humanoid motions in our CoSCo-HSOM algorithm. We have also made progress towards doing reinforcement learning in such structures by successfully demonstrating mechanisms for future reward prediction and hidden state identification in hierarchies of fixed size temporal sequences. The current version of CoSCo-HSOM does not use any feedback from the environment, limiting its capabilities to only creating open loop motions. This preliminary work still cannot fit RoboCup needs, since the reproduced motions are jerky and unstable, but our learning approach is constantly improving.

**E. Obstacle Avoidance and Behavior (TUC)**

The SPL rules postulate severe penalties for player pushing, therefore we have dedicated considerable effort in avoiding collisions. Because of the limited coverage and the noisy readings of the Nao ultrasonic sensors, each robot builds and maintains a local obstacle map of its surrounding. This map is stored in a $7 \times 36$ polar grid (10cm and $10^\circ$ resolution, respectively), which covers a $360^\circ$ area of radius 70cm around the robot. The robot is always located at the center of the grid and the grid moves with the robot. This polar map is updated using distance readings from the ultrasonic sensors. The polar topology facilitates the mapping of the sensor data and also captures the resolution of the sensors which is dense close to the robot and sparse away from the robot due to the conical shape of the beams. Locomotion of the robot implies appropriate transformations on the polar grid. Rotational and
translational motions are reflected on the map in constant time through precomputed and stored transformations, albeit with some loss of accuracy due to discretization. Each cell in the map stores the probability that there is an obstacle at the corresponding position in the field. These values are updated according to a simple sensor model (for cells within the sensor cone) or according to an aging model (for cells outside the sensor cone). The polar obstacle map is used to plan obstacle-free paths in any desired direction of motion or to any desired destination.

The highest level of decision making in our team is modeled as a statechart using a model-based process for agent engineering (ASEME). ASEME enables the designer to specify complex behaviors through a series of semi-automated design phases that produce the final statechart model which is then transformed automatically to source code, compiled, and executed using our multi-threaded statechart engine. The final statechart model specifies the intra- and the inter-agent control for achieving the desired behavior by accessing directly the functionality provided by our base activities: GameController, Vision, Localization, ObstacleAvoidance, MotionController, HeadHandler, Sensors, LedHandler. Using this principled methodology, we redeveloped our simple Goalie and Player behaviors from RoboCup 2010, previously implemented as MONAS sensors, LedHandler. The added value of ASEME was revealed in the development of a team attack protocol involving two players (center and center-for interchangeable roles), which was supported by the ability of ASEME to specify inter-agent control and the ability of MONAS/NARUKOM to realize it on the robots. We are currently working on a more complex protocol that covers several cooperation strategies for the entire team.

V. TEAM PUBLICATIONS

The members of the team regularly publish their research results. The list below includes only publications related directly or indirectly to RoboCup. Other publications may be found on the member group’s web pages.