

# Linköping Humanoids – Team Description

Fredrik Heintz, Fredrik Löfgren, Mattias Tiger, Gustav Häger, Jon Dybeck,  
Karl Holmquist, Tore Haglund, and Michael Felsberg

Department of Computer Science and Department of Electrical Engineering  
Linköping University, Sweden

`fredrik.heintz@liu.se, frelo223@student.liu.se, mattias.tiger@liu.se,`  
`gustav.hager@liu.se, jondy276@student.liu.se, karho935@student.liu.se,`  
`torha599@student.liu.se, michael.felsberg@liu.se`

**Abstract.** A high-level overview of the newly formed RoboCup SPL team **Linköping Humanoids** from Linköping University in Sweden. The team consists of students and faculty from two departments and three study programs. The team focuses on perception and decision making. The perception system aims at being capable of detecting and tracking both the ball and the other robots as well as provide localization of the robot. The decision making system consists of high-level strategies, dynamic role allocation and role-based decision making.

## 1 Introduction

This is the team description of the newly formed RoboCup SPL team **Linköping Humanoids** from Linköping University (LiU) in Sweden. Our team represents the student association FIA Robotics, the Division for Artificial Intelligence and Integrated Computer Systems (AIICS) at the Department of Computer Science (IDA) and the Computer Vision Laboratory (CVL) department at the Department of Electrical Engineering (ISY). The team consists of:

- Fredrik Heintz, Associate Professor AIICS, Team Leader
- Fredrik Löfgren, FIA, student Applied Physics and Electrical Engineering, Team Leader
- Jon Dybeck, FIA, student Computer Science and Engineering
- Mattias Tiger, PhD student AIICS
- Karl, Student FIA, student Applied Physics and Electrical Engineering
- Tore Haglund, FIA, student Cognitive Science
- Gustav Häger, PhD Student CVL
- Michael Felsberg, Professor CVL

We all have different backgrounds, but we all share a common interest in robotics. A major result of this project is the collaboration between the students, PhD students and faculty from two departments and three education programs.

The team acknowledges the generous funding from the Department of Computer Science (IDA), the Division for Artificial Intelligence and Integrated Computer Systems (AIICS), the Computer Vision Laboratory (CVL), the program

board for Electrical Engineering and Physics (EF-nämnden), the National Graduate School in Computer Science, Sweden (CUGS), the Swedish Aeronautics Research Council (NFFP6), the Swedish Foundation for Strategic Research (SSF) project CUAS, the Swedish Research Council (VR) Linnaeus Center CADICS, the ELLIIT Excellence Center at Linköping-Lund for Information Technology, the Center for Industrial Information Technology CENIIT, and VINNOVA, Sweden's innovation agency, project Computational Thinking.

### 1.1 Background on the Team

AIICS has a long history of research in artificial intelligence and its application to intelligent artifacts. Intelligent artifacts are defined as man-made physical systems containing computational equipment and software that provide them with capabilities for receiving and comprehending sensory data, for reasoning and for rational action in their environment. Examples of such artifacts range from PDAs and software agents to ground and aerial robots. An equally important focus is the development of integrated systems which include hardware, software, sensors and human users. AIICS has focused mainly on unmanned aerial vehicles but the techniques and technologies developed can also be applied to humanoid robots. Our research includes for example logic-based spatio-temporal reasoning over streaming data, automated task and motion planning, task allocation in multi agent systems, and localization and navigation of robots.

For more information on research at AIICS see: <http://www.ida.liu.se/divisions/aiics/>.

The research at CVL covers a wide range of topics within artificial visual systems (AVS): three-dimensional computer vision, cognitive vision systems, object recognition, image analysis and medical imaging.

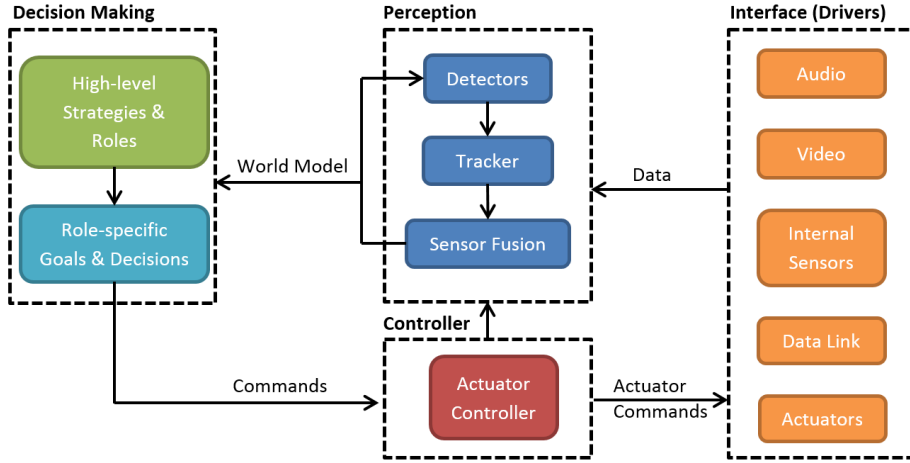
Computers are better than humans at playing chess, but even a small child has better generic vision capabilities, as required for robot vision and playing soccer, than any artificial system. CVL aims at improving AVS capabilities substantially, driven by a human visual system inspired approach, as AVS are supposed to coexist with – and therefore predict actions of – humans.

CVL has more than a decade of research experience on real-time computer vision for robotics and building robotic systems with distributed computations using, e.g. ICE and ROS. CVL's research focuses on systems with layered feedback loops that are formed by online perception-action learning. These systems have been applied on various platforms, such as manipulators, mobile ground vehicles, and unmanned areal vehicles.

For more information on research at CVL see: <http://www.cvl.isy.liu.se/research/>.

## 2 Architecture

An overview of the architecture is shown in Figure 1. The overall system consists of a low-level system responsible for robust perception and control as well as



**Fig. 1.** High-level overview of the architecture.

interfacing with the Nao and a high-level system responsible for decision-making. ROS, the Robot Operating System, is used as the communication infrastructure. To limit the computational overhead there are only two ROS-nodes running in separate processes. The first node is the low-level system which is a Nodelet Manager running the individual perception and control nodes. The second node is the high-level system. The communication between them is done using two topics, one topic where the current world model from the perception system is published and one where the current command to the controller is published. A new decision is made every time a new world model message is received.

The different subsystems are described in more detail below.

### 3 Perception System

The perception system is currently a vision system. The vision system is composed of a number of detectors and a scheduling unit. There are detectors for lines, balls and goals. Each detector operates on only a small region of the image at the time, in order to reduce computational needs. The regions to run a detector in are selected by the scheduler based on the current estimated playing field, as well as what is most important at a particular time (getting a new position estimate, following the ball or finding a goal).

As each patch only covers a small part of the image, the output from a number of detections is combined to form a final detection before being passed to further in the perception pipeline.

The lines are detected by first creating a gradient magnitude image from the grayscale channel of the input patch. The Hough transform is then used on the gradient image after binarization in order to obtain the set of lines present in the patch. The detections from each patch is then combined by first binning

lines of with similar angles in the image plane together. Then each direction bin is further binned by their position in the image, and similar detections are considered to be part of the same line. Corners are estimated from these lines as the positions where they are at right angles. The lines and corners are then passed on to the rest of the perception system.

Balls are detected by estimating the number of pixels within a particular color interval in a region, if sufficient number of pixels have the correct color the region is considered to contain part of a ball. For the ball detector the regions are combined by checking close to the edge of the region, if the edge of the region has contains the ball the detector window will move in that direction until the entire ball has been found, in order to allow an estimation of the ball center.

As the goals have the same color as the lines they tend to be detected as lines, however as they are physically located in a plane orthogonal to the ground any lines detected to far away from the robot to reasonably lie on the ground plane is instead projected onto an estimated goal plane. The goal plane position relative to the ground plane is estimated by detecting the point where the goal supports intersect the ground plane.

In order to aid the attention focusing mechanism there is a statically computed distance map for each pixel row in the image, computed offline for some different angles of the head. This makes it simple to make sure the region of attention is never positioned at some unreasonable distance away from the robot (outside the playing field). With accurate positioning inside the playing field available this will further reduce the number of regions it is worth looking at, as we can then set a maximum distance from the robot that is smaller than the length of the playing field.

### 3.1 Localization

Objects detected in the image plane are projected down onto the soccer field plane using similarity-transformations of the robot joints and calibrated camera parameters. The projection error of each object is captured by projecting down a circle surrounding the object detection in the image plane onto the soccer field plane. The circle serves as a crude estimation of the uncertainty associated with the point estimate of the position of the object detection and the circle's diameter is proportional to the size of the object detection in the image plane.

Each robot has a qualitative map of the soccer field in terms of line features such as lines and line crossings. Each robot localizes itself in the world using these detected line features by comparing them with the qualitative map. The qualitative world map is a static map representing the soccer field and containing the positions and relations of soccer field lines, line crossings and goal posts. RANSAC is used to make the matching robust against outliers and noise. Usually a robot only sees a small part of the soccer field and therefore do not see enough line features to precisely localize itself. When this is the case, the robot generates a set of possible worlds. This set of multiple hypotheses is reduced to a single location by checking the expected line features for the hypothetical worlds and discarding those that do not correspond to the observed reality. The localization

of the robot is updated every now and then while walking which makes the drift from the odometry manageable. The difficult part is the initialization of the first localization when no other robot can provide a good estimate over wifi already.

Observed objects are positioned in the robots mental map of the soccer field and their dynamics are updated using a Kalman filter per object, regardless if they are in view or not. Static objects only have a position, robots have a position and a velocity, and the ball has a position, a velocity and an acceleration where the acceleration is used to predict where a moving ball will come to a halt by itself. All static objects (goal and line features) will be indirectly measured by observing any one of them. When one of the static objects is observed, all other are updated based on that observation since the static objects have known relative positions and rotations to each other.

A multi-hypothesis tracker is used to maintain assigned object identities throughout the game. This is important in the short term of game play where the Kalman filter states need to be object specific in order to provide meaningful state estimations. The multiple hypothesis part makes the system more robust against false detections which have not been pruned away. The tracker is a greedy first algorithm which minimizes the distance between objects in position while taking the projection error into account.

The robots mental map of the world which includes the expected location of objects is used as prior knowledge by the detectors. This means that the vision system can first and foremost be focused on regions of interest around certain objects which is regions that are smaller than the entire image. This reduce the amount of image processing required for object detection. Since the robot has a crude estimate of where objects ought to be on the soccer field even when not seeing them, it also has prior knowledge of where to start looking for objects in the world (i.e. in order of getting more certain positions of them when they have been out of view for a short while).

## 4 Controller

The control of the robot is quite rudimentary. We are using the standard walking engine but have implemented a small number of control modes such as kicking in different directions, sitting down to protect the goal and walking side-ways.

## 5 Decision Making

The overall strategy of our team is to reduce the effect of our main weakness, which is the speed, by leveraging our expected strength which is the perception (vision and localization system) system and the fact that we can kick in many directions without repositioning. Since we fear that we will be one of the weaker teams, we will focus on the defensive. Finally we focus on realizing a simple and hopefully robust decision making mechanism.

The general decision procedure of each agent is:

1. Determine the *strategy* based on facts from the game controller.
2. Determine my *role*. The strategy determines the roles.
3. Determine *sub-roles*. The role determines the sub-roles.
4. Determine the *goal*. The role determines the goals.
5. Make a decision by generating options, evaluating them and then selecting the best one.
6. Send a new *command* to the controller if necessary.

### 5.1 Strategies and Roles

There are three overall strategies Offensive, Offensive-Defensive and Defensive. The strategy selected depends on the half, the remaining time in the half, the current score difference and how many points we are aiming for in the game (1 or 3). This can easily be configured. The general strategy is: When in the lead be defensive. When behind be offensive. When a draw be offensive-defensive, unless we need to win then offensive or we must not lose then defensive.

Each strategy defines a set of roles. A robot has exactly one role. To handle disabled or broken robots the role assignment is dynamic based on a priority for each role and the cost for switching roles. The priority is a fixed value plus a dynamic value which depends on the time until the original robot comes back (a broken robot is never expected back). The cost depends on the priority of the current role and the distance to the desired position for the new role.

Each role defines a set of sub-roles and goals. A sub-role is basically a role within the role, but where a robot has exactly one role it may have zero or more sub-roles. They are used to divide the problem of making a decision into smaller parts and to provide information to the other robots about the robot's intentions. Based on the sub-role and the current world model the robot makes a decision about which command to execute. If the command is already being executed nothing is done.

## 6 Summary

This paper provides a high-level overview of the RoboCup SPL team **Linköping Humanoids** from Linköping University in Sweden. The team consists of students and faculty from two departments and three study programs. The team started basically from scratch and developed the whole system in less than six months where most of the work was done the last two months. The main goals of the team is to do well enough to qualify for RoboCup 2016 and to learn as much as possible about developing a successful RoboCup SPL team.