Nao Devils Dortmund

Team Description for RoboCup 2015

Matthias Hofmann, Ingmar Schwarz, and Oliver Urbann

Robotics Research Institute
Section Information Technology
TU Dortmund University
44221 Dortmund, Germany

1 Introduction

The *Nao Devils Dortmund* are a RoboCup team by the Robotics Research Institute of TU Dortmund University participating in the Nao Standard Platform League since 2009 and in 2008 as part of team *BreDoBrothers*. A more comprehensive report about the team's research activities up to 2014 is published in form of a team report available online[1].

2 RoboCup Achievements

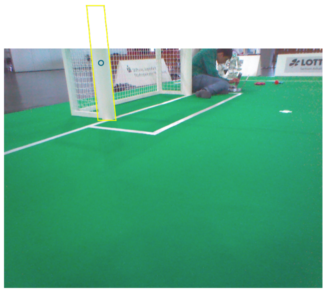
The *Nao Devils Dortmund* have their roots in the teams *Microsoft Hellhounds* (and therefore part of the *German Team*), *DoH!Bots* and *BreDoBrothers*. The team had a number of successes, such as winning the RoboCup World Championship twice with the *GermanTeam* (2004 and 2005), winning the RoboCup German Open 2005, the Dutch Open and US Open 2006 with the *Microsoft Hellhounds*, and winning the Four-Legged League Technical Challenge two times (2003 by the GermanTeam, 2006 by the *Microsoft Hellhounds*). In parallel to these activities, the *BreDoBrothers* started a joint team of TU Dortmund University and University Bremen in the Humanoid League which participated in RoboCup 2006. The *DoH Bots!* designed and constructed a humanoid robot from scratch and competed in the Humanoid League of RoboCup 2007. Team *BreDoBrothers* participated successfully in the first Nao Standard Platform league in 2008 when it reached the quarter finals.

The *Nao Devils Dortmund* were founded in 2008 and placed 3rd out of 9 teams in the German Open 2009, 3rd out of 24 teams in the RoboCup 2009, 2nd out of 27 teams in the RoboCup 2011, and 3rd out of 14 teams in the German Open 2012 and reached the third place of the technical challenges in 2013. Since 2011, our team was prequalified every year, and we reached at least the quarter finals. In 2014, we reached the 3rd place in the Drop-In competition, and the 4th place in the technical challenges. In 2015, the robots of team Nao Devils will play with customized jerseys.

3 Research

The cooperative and competitive nature of robot soccer in the Standard Platform League provides a suitable test bed for a broad research area. Thus, *Nao Devils*' research is mainly

[1] http://nao-devils.de/wp-content/uploads/2014/12/NDDTeamReport2014.pdf

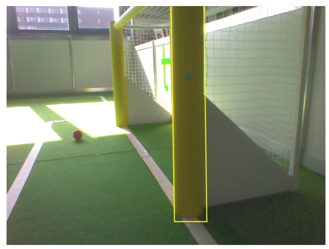


(a) White goal detection at the German Open 2015.

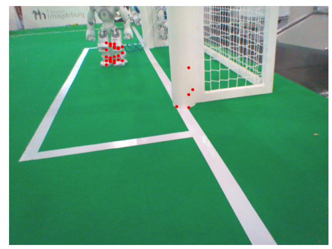
(b) Goal post base point adjusted with line detection.

(c) Correctly detected goal posts with side indication.

(d) White goal detection error due to similar background.



(e) Detection of a yellow goal post.



(f) 1D-Features on robot and goal post.

Fig. 1. White goal detection and robot detection results in different conditions - no calibration was done throughout presented images.

focused on computer vision, localisation, artificial intelligence, and humanoid walking. Due to rule changes, the major challenges in 2015 are the robustness of the goal detection, and changes in the localization system to improve the independence against false positives.

3.1 Calibration-free Image Processing

In comparison to 2014, the obvious challenge was to detect white goals. To this end, the current concept was enhanced to not only use yellow features but edges in general as the input for goal perception. These edges are then connected to goal post side lines and scans are performed to verify width, height and unity of color on the goal post as in the previous years. As an additional measure to identify the position of the goal post on the field, detected field lines are used since there might not be a visible edge between field line and the bottom of the goal post.

The white goals presented a problem for our robot detection since depending on the structure of the goal it contains a lot of white resulting in false positive percepts in the goal area. This problem was solved by adding a simple 1D-Feature scan on the legs of the possible robot. On a real robot a lot more of these features are present making it easy to distinguish between a real robot and a false positive percept.

Figure 1 shows some examples of the white goal detection and an example of found 1D-Features on a robot. The current implementation allows detecting all goal posts independent on their color, as long as the edges are found.

3.2 Preview Kick

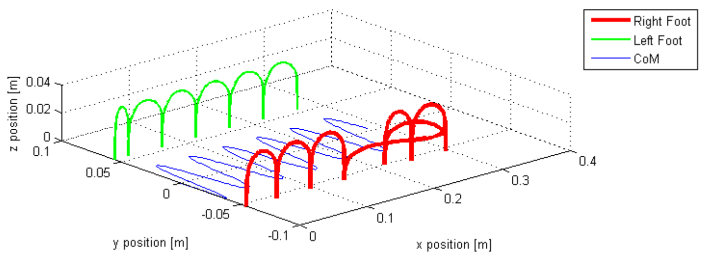


Fig. 2. Feet positions in world coordinate system of a kick with the right foot during a walk with $\dot{x} = 5cm/sec$. The ball lays at $y = -5.5cm$ and is kicked to the left side. The kick is executed within the walk without the need to stand before or after the kick.

In previous years we applied a kick that is executed within the swing phase during the walking motion without changes. It is a fast and robust kick but due to the requirement of the walking engine to provide a preview of the ZMP, the position of the feet cannot be altered immediately and the kick must start and end at the preplanned foot positions, see Fig. 2. Thus, the foot must be moved to the ball and back to the desired position, resulting in a large distance for a short time that reduces the precision of the executed foot motion.

We therefore enhanced the structure of the kick that it is planned approximately half a second before it is executed. Thus, we are able to reduce the length by placing the foot on the position of the ball. This way the kick is executed with only approximately half of the distance and a significantly improved precision.

3.3 Rule-based Path Planing

The Dortmund WalkingEngine accepts as input the desired speed which can be directly set by the behavior. However, usually it is important to avoid obstacles like other robots or goal posts to reach the desired target which makes it difficult to set the speed based on a state machine. Path planing aims at creating in path beginning at the own position to the desired target to simplify this task. Nevertheless, the path must be converted to the desired speed as this is the only input accepted by the Dortmund WalkingEngine. This is a critical and often underestimated issue as the Dortmund WalkingEngine is fully omni-directional and walking along a path with a desired orientation has more than one solution. The conversion from the desired path to the desired speed therefore includes also decision that influence the overall time till the target is reached. As a result, we developed a module that focus on this conversion with a high priority on rapid development and efficient debugging.

The conversion is based on rules as depicted in Fig. 3. A rule has the following properties:

– A number of entry conditions that are all must be true to enter into this rule, robot3.PathToSpeed

ID	Value	Entry Min	Entry Max	Exit Min	Exit Max	Factor X	ID X	Value	Factor Y	ID Y	Value	Factor R	ID R	Value
Next_WP_Angle	0.827432	50	180	-20	70	0	Max_Forward	300	0	Next_WP_Angle	0.82743	1	Max_R	45.8366
Next_WP_Dist	402.329	300	10000	-100	500	0	Max_Forward	300	0	Next_WP_Angle	0.82743	-1	Max_R	45.8366
Next_WP_Angle	0.827432	-180	-50	-50	30	0	Max_Forward	300	0	Next_WP_Angle	0.82743	-1	Max_R	45.8366
Next_WP_Dist	402.329	2000	10000	-300	200	1	Max_Forward	300	0	Max_Left	140	0	Max_R	45.8366
Next_WP_Dist	402.329	600	2000	-300	500	0.8	Max_Forward	300	0	Next_WP_Dist_Y	5.81	0.4	Next_WP_Angle	0.827432
Next_WP_Dist_Y	5.81	0	10	-10	200	150	One	1	1.5	Next_WP_Dist_Y	5.81	1.5	Next_WP_Angle	0.827432
Next_WP_Angle	0.827432	0	8	-20	20	150	One	1	1.5	Next_WP_Dist_Y	5.81	1.5	Next_WP_Angle	0.827432
Angle_at_Next_W	0.533875	0	8	-20	20	150	One	1	1.5	Next_WP_Dist_Y	5.81	1.5	Next_WP_Angle	0.827432
						1.4	Next_WP_Dist_X	402.287	2.6	Next_WP_Dist_Y	5.81	3	Angle_at_Next_WF	0.533875

Fig. 3. Window with rules during normal behaviour.

– a number of exit conditions that define a hysteresis before the rule can be leaved,
– for all dimensions (x, y and rotation) a constant factor with a multiplier that defines the speed that is executed as long as this rule is active.

All conditions are define by a minimum value and a maximum value that are compared to a variable. If all variables are within the desired range, the rule is selected. To exit a rule, all variables must be outside the range that is define by the exit minimum and maximum value. Many variables are available:

– Euclidean distance to next way point,
– angle to the next way point,
– maximum speed in x and y direction and maximum rotational speed,
– angle to the destination,
– distance to destination,
– angle at next way point (relative to current angle),
– distances to next way point on single axes (x and y),
– distance to next obstacle,
– angle to the ball,
– distance to the ball,
– and the constant 1.

Some values are only useful for definition of the resulting speed. E.g. the maximum speed can be multiplied with an arbitrary value to select a constant speed.

All rules are prioritized. If a rule with higher priority (higher position in the hierarchy as depicted in Fig. 3) and the entry conditions are met, the exit conditions of the previously selected rule with a lower priority are ignored and the higher rule is selected immediately.

For fast debugging we developed the GUI as shown in Fig. 3. The colors clarify which rules is active and why (green: entry conditions met, red: exit conditions not met). Also, the current values are printed. Additionally, the rules can be changed online and reloaded during execution in simulation as well as on a physical robot.

3.4 Motion Control

Motion generation can be divided into periodic motions, such as walking, and non-periodic motions, such as kicking. To define periodic motions, our closed-loop approach focuses on the use of different sensors to measure the stability of the executed motion. A path provider calculates a reasonable path to the destination using a potential field as shown in figure **??**. It passes the necessary speed and direction on to a pattern generator in order to follow the path. Subsequently, the pattern generator forms suitable footstep positions to reach the desired walking motion. To generate the robot motions an inverted-pendulum model is used to generate gait walking patterns. A stable execution of the patterns is ensured by means of ZMP measurement and an appropriate preview controller [1, 2].

Since motions on real humanoid robots reveal instabilities caused by inaccuracies of the used servos and external disturbances an observer is utilized to measure the actual state of the robot. Since 2011, we apply a sensor fusion approach [6]. The main source to compute the actual ZMP are the Force Resistant Sensors in the feet in addition to the measured angles. The result of the controller and the state is a damped reaction to disturbances such that self-induced oscillations are avoided.

Besides this sensor feedback, two other heuristics are implemented to further stabilize the walk. The gyroscopes are employed to directly control the body orientation. This has also a dampening effect. Similarly, way the acceleration sensor is utilized to modify the x position of the body.

The shown approach to generate walking motions has proven successful during RoboCup 2008 and has been further improved and extended resulting in stable walking speeds up to $44\,cm/sec$ during RoboCup 2010.

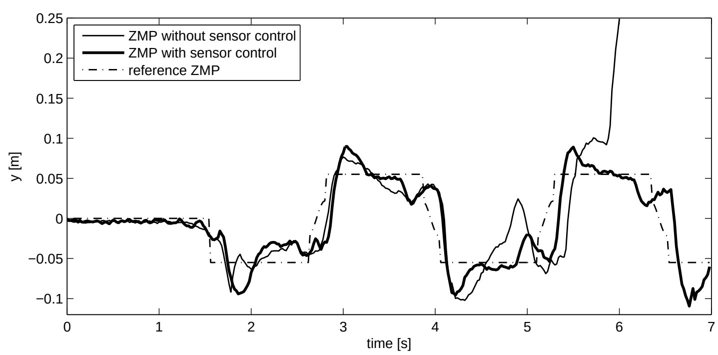


Fig. 4. The effect of sensor feedback control on a walking motion that was not calibrated for a real robot but for a simulation model. Without sensor control the real robot falls after a few steps, while with sensor control it is capable of compensating the differences of the internal model from the real robot's mechanical and physical properties.

Applying sensor feedback to supervise robot stability during execution of predefined motions would lead to more stability. Hence, approaches to observe the execution of predefined motions by means of a controller are a research focus of team *Nao Devils* [3]. As a result, the kicking motion is integrated into the walking controller. Consequently, kicking is no longer a separate predefined motion, and it is not necessary to stop the robot before kicking. The kicking motion starts right after the last walking step and the walk continues without standing as shown in figure 2. The kick direction can be chosen by the behavior at the start of the kick.

Even if the stabilizing effect can be shown, it is obvious that not every disturbance can be balanced this way. From observing human beings it can be followed that large disturbances can only be balanced by modifying the desired foot placement. This is also true for walking robots, but the derivation is different. In case of a preview controller that balances the measured difference between the estimated state and the desired state, the motivation for lunges is to modify the reference ZMP such that the closed-loop systems is controlled like in the open-loop case. Modifying the reference that way mitigates the error between the measured ZMP and the desired almost entirely. Thus, large disturbances are easier to handle.

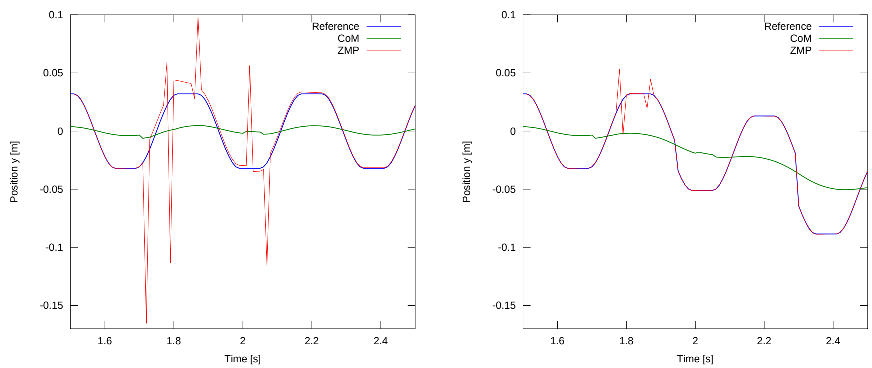


Fig. 5. Balancing without lunges (on the left) and with lunges (on the right).

It can be shown that applying the above mentioned requirement to the equation system of the preview controller/observer leads to a matrix that can be used to calculate the reference ZMP modification in closed-form [5]. Figure 5 shows an example walk of a robot simulated using the 3D linear inverted pendulum mode. At time $\sim 1.7s$, $\sim 2s$ and $\sim 2.1s$ center of mass errors are measured, and at time $\sim 1.8s$ and $\sim 1.9s$ ZMP errors. As can be seen, without lunges the balancing leads to further deviations in the ZMP while lunges minimize the errors. Details about the derivation and reasons for the deviations are given in [5].

Besides the question about the mathematical realization of the step modification, the implementation of the robot must also consider various topics to realize modification that lead to a stabilization. A tilting robot must be raised after a lunge which requires high torques. We therefore lower intentionally the CoM linear to the measured body tilt. Additionally, the swinging foot is rotated around the stand leg to avoid undesired collisions with the ground.

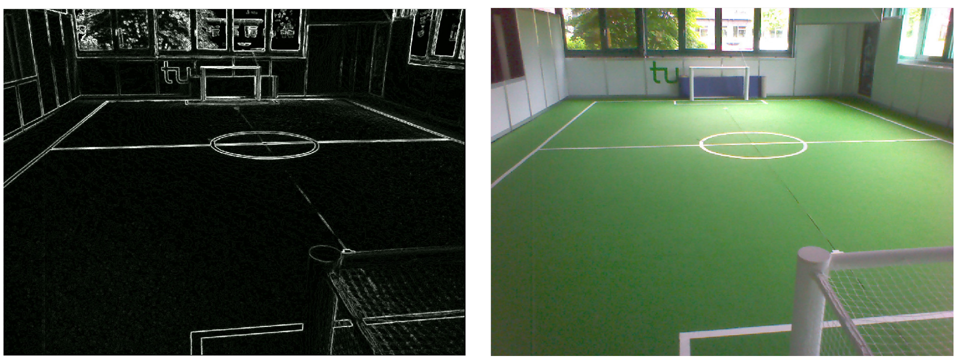
While it can be proven in various experiments that this modification can stabilize the robot while it would fall down without, we currently investigate the advantage while walking with high speeds. However, the modification is clearly needed to be able to walk on problematic floors, like artificial grass etc.

3.5 Coach

The coach robot has been introduced into the SPL in 2014. Its purpose is to observe the game and potentially provide guidance to the team by giving tactical advice. However, only a very few teams put effort into developing a coach robot. This is mainly due to the following reasons: First, the rules were very restrictive, e.g. messages the coach sent out to its team were significantly delayed. Second, albeit a some code can be reused from the players, teams probably have to develop specialized modules for the coach robot. Team Nao Devils decided to implement a coach robot for this year's main competition.

Since the coach is sitting on a table, and observes the field, and is not allowed to receive any packages from the field players, the coach has to detect reliably robots, the ball, and the field. Hence, we designed a new image processing system for the coach. Apart from the fact that the implementation is work in progress, we show first result in this report:

In order to detect field lines, and goals, we successfully implemented a high-performance, SSE-based Sobel edge detector (see Figure 6). SSE is a special instruction set extension designed by Intel. The SSE sobel runs on the Nao in average below 15ms on the HD picture provided by the Nao camera.

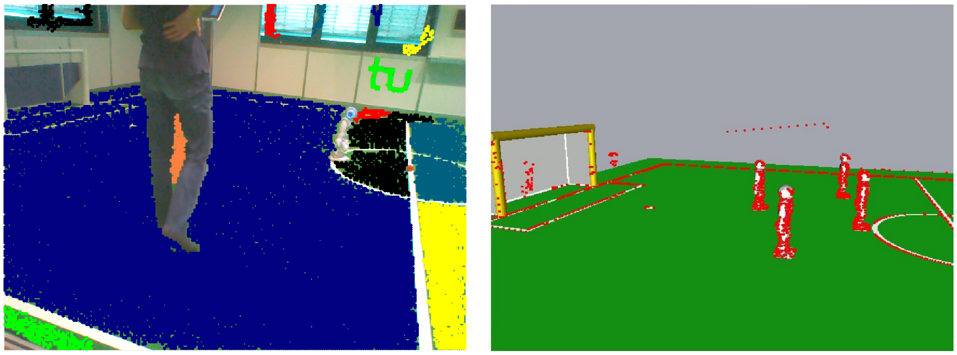


(a) Sobel image. (b) Original image.

Fig. 6. Sobel-based image processing.

Regarding the field and robot detection, we utilize the HSI color space. Figure 7 shows on the left side a segmentation of the field. There are different parts (however, still with some errors and problems). The overall goal is to build a convex hull around the field. On the other hand, we use a clustering-based method [4] to identify points where potential robots could be present on the field. The next step is the ability to distinguish between own and opponent robots, and robot tracking.

Since the coach is going to give tactical advise to the players, it must be capable to detect game situations based on top of the image processor. This kind of meta-information extraction is challenging since it is even hard for humans to detect the nature of the situation. We conducted a first experiment where we generated 100 game situations. Experts were asked to classify the game situation into categories like 'defense', and 'offense'. We are now working on the generation of rules according to the knowledge given.



(a) Segmentation of the field.

(b) Robot detection.

Fig. 7. .

4 Conclusion and Future Work

This report summarizes the developments in 2015 by team Nao Devils to improve existing soccer playing abilities in the SPL. Generally, a lot of effort has been put into the robustness of the system. Our focus this year lies on the implementation of the changes imposed by the rule book. Especially the introduction of white goals caused changes in the way how we detect goal posts, and how we use the information in the localization system. Moreover, we have shown improvements regarding our kick, and described our activities regarding the coach. However, the strategical usefulness of the coach has to be proven when the fundamentals have been laid.

After reaching a certain level of maturity of basic skills needed to play soccer, future work will focus on cooperative team behaviors, esp. dynamic and situation-dependent decision making (positioning).

References

1. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57(8), 839 – 845 (2009), humanoid Soccer Robots
2. Czarnetzki, S., Kerner, S., Urbann, O.: Applying dynamic walking control for biped robots. In: RoboCup 2009: Robot Soccer World Cup XIII. pp. 69 – 80. Lecture Notes in Artificial Intelligence, Springer (2010)
3. Czarnetzki, S., Kerner, S., Klagges, D.: Combining key frame based motion design with controlled movement execution. In: RoboCup 2009: Robot Soccer World Cup XIII. pp. 58–68. Lecture Notes in Artificial Intelligence, Springer (2010)
4. Rosten, E., Porter, R., Drummond, T.: Faster and better: A machine learning approach to corner detection. IEEE Trans. Pattern Anal. Mach. Intell. 32(1), 105–119 (Jan 2010)
5. Urbann, O., Hofmann, M.: Modification of foot placement for balancing using a preview controller based humanoid walking algorithm. In: RoboCup 2013: Robot Soccer World Cup XVII. Lecture Notes in Artificial Intelligence, Springer (2014), to appear
6. Urbann, O., Tasse, S.: Observer based biped walking control, a sensor fusion approach. Autonomous Robots pp. 1–13 (2013), http://dx.doi.org/10.1007/s10514-013-9333-4