Nao Devils Dortmund

Team Description Paper for RoboCup 2017

Matthias Hofmann, Ingmar Schwarz, and Oliver Urbann

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

1 Team

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*.

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. Since 2011, our team was pre-qualified every year, and reached at least the quarter finals, or intermediate rounds. In 2016, Nao Devils won the Outdoor competition, and the Technical challenge competition. In 2017, we recently achieved the 3rd place in the German Open. According to its prior rankings, team Nao Devils will be seeded into the Champions Cup in 2017. Team members are mainly master and PhD students.

A more comprehensive report about the team's research activities up to 2016 is published in form of a team report available online[1].

2 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 focused on computer vision, localization, artificial intelligence, and humanoid walking.

[1] https://github.com/NaoDevils/CodeRelease2016/blob/master/TeamReportNaoDevils2016.pdf

Due to rule changes, the major challenges in 2017 are the artificial carpet, and the natural lighting which is still a major challenge in RoboCup.

There have been major scientific contributions of the team during the past years. Hence, this team description paper outlines two subjects that will be applied first time in RoboCup 2017 by our team.

2.1 Deep Learning

Convolutional neural networks (CNN) are a state-of-the-art approach for detecting objects in images. Due to its high success rates, and the presence of very few false positives, more and more teams are using this methodology for detecting the ball, robots, or other objects [6][1]. A major drawback of CNN is that they require comparatively much computational power. In RoboCup, it is desirable to meet real-time conditions[2]. Unfortunately, due to the limited hardware of the NAO, this requirement if often hard to achieve. Therefore, we implemented a CNN with special focus on minimizing computational power.

The process of our ball detection is as follows: First, we create various ball hypotheses per frame in preprocessing step. Second, all ball hypotheses are extracted and scaled to 16x16 pixels. Since we are exclusively using the luminance channel of the images, the 16x16 grayscale images serves as an input for the CNN. The task of the CNN to check whether the hypothesis is a ball or not. The structure of the CNN layer is the following:

1. Convolution 6x6x1, Stride 2, Pad 2, 4 Filter
2. Max-Pooling 3x3, Stride 3, Pad 1
3. Convolution 3x3x4, Stride 2, Pad 1, 12 Filter
4. Max-Pooling 2x2, Stride 2, Pad 1
5. Rectifying Linear Unit
6. Convolution 2x2x12, Stride 2, Pad 0, 2 Filter

We use the MATLAB toolbox MatConvNet[3] for training and validation. The learning method is back propagation. The test samples are generated from log data of the robot. The log data has been recorded in various lighting conditions, and heterogeneous game situations. Moreover, different distances of the samples to the perceiving robot are used. To enlarge the amount of images used for training, the positive samples of the ball are rotated by 90, 180, and 270 degrees. Thus, our image database is about 300,000 samples.

After training the network, we generate C++ code that does not consist any loops, control structures, or function calls. This is done by a script that runs in MATLAB. The generated C++ code is then used in our robotic framework. Since major parts of the code utilize SSE2 optimizations, one ball hypothesis can be classified within 45μs. The generated code, however, consists of about 18500 lines.

Figures 1 and 2 are views from the goalie in the 3rd place game of the German Open 2017. The green circles indicate ball hypotheses. They are generated from preprocessing in our vision system [5]. The blue circle is a validated ball percept, and the colorings on the ball indicates that the ball is part of the ball model. The ball model requires a number of percepts to be accepted as ball. This is in order to be less susceptible against false positives. Checking the log data suggests that there are very few false positives, but they are below a critical mass that would affect the ball model.

---

[2] We define real-time capability by computing 60 images per second from the cameras of the NAO.

[3] http://www.vlfeat.org/matconvnet/

Fig. 1. Sample image with 11 ball hypotheses.

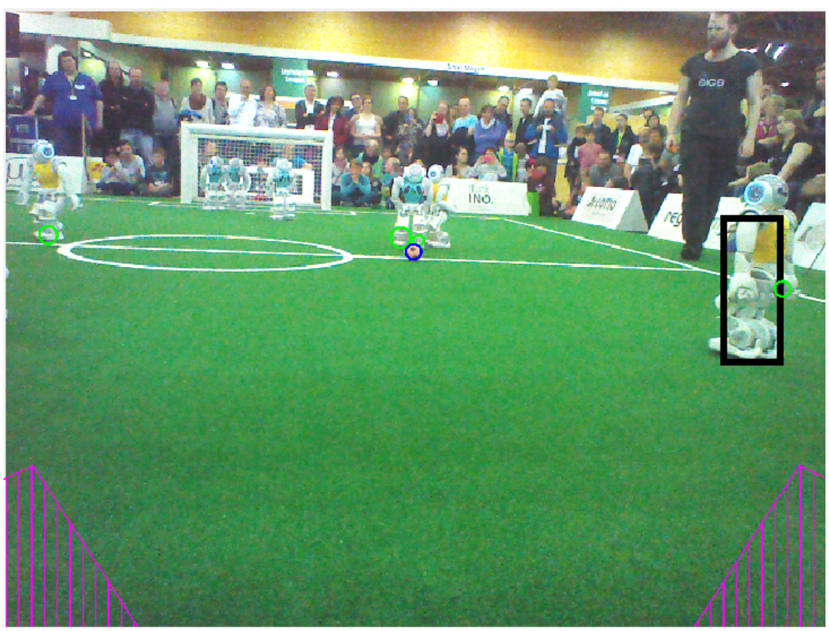

Fig. 2. The robot is able to detect a ball from approximately 4.5 meters distance.

2.2 Motion Control

Flexible Linear Inverted Pendulum Model: For the upcoming RoboCup 2017, we use the FLIP model which we recently introduced [7]. This entire section is based on the findings, and description of the paper. We derived a discrete state-space representation that can be applied by a preview controller as proposed by Kajita et al. [3][4].

The well-known LIPM proposed by Kajita et al. [4] is a simplification of the robot dynamics to a single center of mass and is usually applied to derive the Zero Moment Point (ZMP)[5] from its dynamics. The ZMP p of this system can be calculated by:

$$p = c_1 + \ddot{c}_1 \cdot \frac{z_h}{g}, \tag{1}$$

where g is the gravity and c_1 the position of the CoM. The linearity is a consequence of the constant height of the CoM and important to design closed-form algorithms for motion generation.

We extended the system by a spring and a damper. To retain linearity we added a second cart with an additional small mass as depicted in Fig. 3. Here, two carts are connected via a damper and a spring (with constants b and k respectively). The left cart (cart 1) has no mass but is connected to CoM 1 with constant height z_h by a pole of variable length. Similarly to the variable length of the pole, the length of the cart is also variable. It is adapted such that the spring and the damper is always connected to the cart at the position c_1 of CoM 1. The right cart (cart 2) has the additional CoM 2 and is the only cart that is accelerated by u. The ZMP of cart 1 is located at p and CoM 1 and 2 at c_1 and c_2 respectively.

FLIPM represents all flexible parts that are accelerated by the motors, e.g. the gears. A spring and a damper connects it to the cart with the large CoM that is only accelerated by the force exerted by the spring and the damper. To control the system only the second cart is actively accelerated. More deriviations, and mathematical details of the work can be found in [7].

As an example, the behavior of the system is depicted in Fig. 4. It consists of a small mass (0.1 kg) and a large mass (1 kg) that are connected by a spring with $k = 1$ and a damper with $b = 0.1$. At time $t = 0.5\text{s}$ an acceleration of $u = 100\frac{\text{m}}{\text{s}^2}$ is applied for one time frame ($T = 0.01\text{s}$).

Preview Controller: To derive a controller that is able to generate the walking motion, we chose to define the walk by a reference trajectory of the ZMP as proposed by Kajita et al. [3]. We therefore define the output of the system as:

$$p_k = \mathbf{c} \cdot \mathbf{x}_k, \tag{2}$$

$$\mathbf{c} = \left(1, 0, -\frac{z_h}{g}, 0, 0, 0\right). \tag{3}$$

[4] A script containing all derived equations and the preview controller can be downloaded here: https://github.com/OliverUrbann/FLIPM/

[5] The Zero Moment Point is a point derived from the dynamics of the robot where all tipping moments are zero. Thus, the robot is assumed to be stable if this point is inside the support polygon, which is the convex hull of all contact points of the robot with the ground.

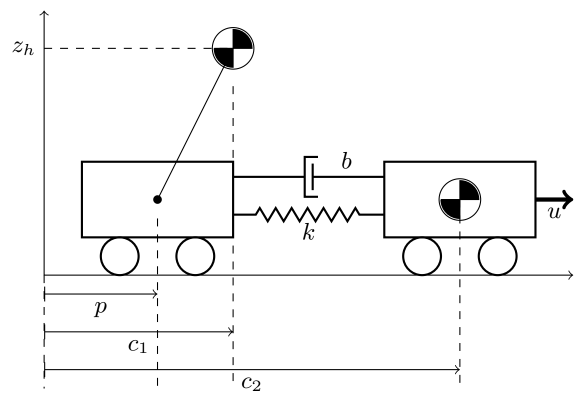

Fig. 3. Flexible Linear Inverted Pendulum Model (FLIPM).

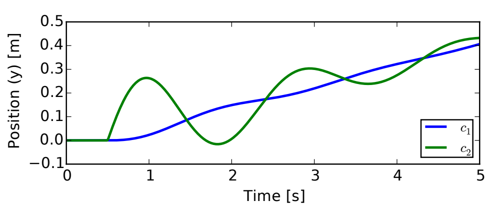

Fig. 4. Example run of the FLIP model by executing the state-space representation for 5 simulated seconds. Mass 1 has a weight of 1 kg, mass 2 of 0.1 kg.

As described in [2] it is not sufficient to define the desired ZMP of the current time frame. A preview of approximately 1s is required. The controller u_k is given by:

$$u_k = -\mathrm{G}_I \sum_{i=0}^{k} \left[\mathbf{C}\mathbf{x}_i - p_i^{ref}\right] - \mathbf{G}_x \mathbf{x}_k - \sum_{j=1}^{N} \mathrm{G}_{d,j} p_{k+j}^{ref}, \tag{4}$$

where the gains $\mathrm{G}_I, \mathbf{G}_x$ and $\mathrm{G}_{d,j}$ that minimize a given cost function are derived by applying the procedure proposed in [8]. It includes the solution of a discrete matrix Riccati equation that can be done offline before the walk is generated online on the robot.

On platforms like the NAO, the input of the joints are the desired joint angles. They can be calculated by applying an inverse kinematic on the desired foot positions f_R given in a local coordinate system of the robot. They are given by the following equation:

$$f_R = c_b - c_2 + f_W, \tag{5}$$

where f_W are the desired foot positions in world coordinate system and c_b the actual CoM position of the robot in the local coordinate system. We determine f_W together with the reference ZMP and calculate c_b by measuring the current joint angles and apply a forward kinematic to combine c_b from all CoMs of the links. As can be seen in Eq. 5, it is not possible to apply the controller output directly to the robot, which is the third derivative of c_2. For further details please see [8]. Assuming the physical parameters (k, b, m_1 and m_2) are selected correctly, the CoM of the physical robot follows the large mass of the model.

3 Conclusion and Future Work

For years, the Nao Devils are working towards more robust systems in RoboCup. This essential due to rule changes, and more realistic game play with natural lighting conditions, and artificial turf. In this outline, we have shown a very efficient implementation of convolutional neural networks that we use for ball detection. Moreover, we have coped an extension of the LIPM model that is being used in the walking engine. Future work will mainly consist of improvements in the behavior of the robots.

References

1. Albani, D., Youssef, A., Suriani, V., Nardi, D., Bloisi, D.: A deep learning approach for object recognition with nao soccer robots. In: Robocup Symposium: Poster presentation (2016)
2. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. Robotics and Autonomous Systems 57(8), 839–845 (2009)
3. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: ICRA. pp. 1620–1626. IEEE (2003)
4. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by a simple three-dimensional inverted pendulum model. Advanced Robotics 17(2), 131–147 (2003)
5. Schwarz, I., Hofmann, M., Urbann, O., Tasse, S.: A robust and calibration-free vision system for humanoid soccer robots. In: Robot Soccer World Cup. pp. 239–250. Springer (2015)

6. Speck, D., Barros, P., Weber, C., Wermter, S.: Ball localization for robocup soccer using convolutional neural networks. In: Proceedings of the 2016 RoboCup International Symposium (RoboCup?2016) (2016)
7. Urbann, O., Schwarz, I., Hofmann, M.: Flexible linear inverted pendulum model for cost-effective biped robots. In: 15th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2015, Seoul, South Korea, November 3-5, 2015. pp. 128–131 (2015), http://dx.doi.org/10.1109/HUMANOIDS.2015.7363525
8. Urbann, O., Tasse, S.: Observer based biped walking control, a sensor fusion approach. Autonomous Robots 35(1), 37–49 (2013)