

RoboCup 2019 AdultSize Winner NimbRo: Deep Learning Perception, In-Walk Kick, Push Recovery, and Team Play Capabilities

Diego Rodriguez, Hafez Farazi, Grzegorz Ficht, Dmytro Pavlichenko, André Brandenburger, Mojtaba Hosseini, Oleg Kosenko, Michael Schreiber, Marcel Missura, and Sven Behnke

Autonomous Intelligent Systems, Computer Science, Univ. of Bonn, Germany
<http://ais.uni-bonn.de> rodriguez@ais.uni-bonn.de,

Abstract. Individual and team capabilities are challenged every year by rule changes and the increasing performance of the soccer teams at RoboCup Humanoid League. For RoboCup 2019 in the AdultSize class, the number of players (2 vs. 2 games) and the field dimensions were increased, which demanded for team coordination and robust visual perception and localization modules. In this paper, we present the latest developments that lead team NimbRo to win the soccer tournament, drop-in games, technical challenges and the Best Humanoid Award of the RoboCup Humanoid League 2019 in Sydney. These developments include a deep learning vision system, in-walk kicks, step-based push-recovery, and team play strategies.

1 Introduction

The Humanoid League contributes to the goal of beating the human soccer world champion by 2050 by gradually making the game rules more FIFA alike. Additionally, individual and team skills are also encouraged by a set of technical challenges. This year, for the AdultSize class, the teams were allowed to



Fig. 1. Humanoid AdultSize team NimbRo at RoboCup 2019 in Sydney.

be composed by two team players. Correspondingly, the field dimensions were updated to 14×9 m. These modifications pose several challenges in terms of perception (further away balls and goalposts to be detected), locomotion (longer distances demanding for a faster gait), localization (robust line detection and state estimation), and team play (coordination between players). This paper presents our recent developments to address these modifications and shows their performance in the competition. These developments include in-walk kicks, a step-based push recovery approach, a vision system based on deep learning and team play strategies.

Our robots won all AdultSize 2019 competitions, namely the soccer tournament, the drop-in games and the technical challenges. Additionally, the NimbRo team was given with Best Humanoid award of the Humanoid League. In RoboCup 2019, we used our fully open-source 3D printed humanoid platform NimbRo-OP2(X) [1,2], shown in Fig. 1 with our human team members. We released a video of the 2019 competition highlights¹.

2 Robot Platforms

During the competition three different robots have been used — NimbRo-OP2 (Fig. 1 first from right), NimbRo-OP2X (Fig. 1 first from left) and Copedo (Fig. 1 second from left). These platforms lead team NimbRo to win all possible competitions last year in the AdultSize League of RoboCup 2018 [3]. Despite the visible differences in the kinematic structure and outer appearance, there is a fundamental level of similarity between the platforms. The joints of all robots are actuated with Robotis Dynamixel actuators. These are controlled through a Robotis CM740 microcontroller board, which also incorporates an IMU with a 3-axis gyroscope and accelerometer. For visual perception, a Logitech C905 USB camera in combination with a wide-angle lens was used.

NimbRo-OP2(X). NimbRo-OP2 [1] and NimbRo-OP2X [2] are our self-developed humanoid robots, where both the hardware² and software³ components are completely open-source. Although the platforms share a similar design and name, there is a number of differences between them. With the same height of 1.34 m, the robots place on the lower end of the AdultSize class requirements. Their 3D printed plastic structure mainly contributes to the low weights of 17.5 kg (OP2) and 19 kg (OP2X). Both robots share a similar joint layout with 18 Degrees of Freedom (DoF), with 5 DoF in a parallel kinematics arrangement per leg, 3 DoF per arm, and 2 DoF actuating the head. A Shuttle X1 Gaming computer with an Nvidia GTX 1060 and an Intel Core i7-7700HQ CPU was mounted inside the hollow trunk of the NimbRo-OP2. For the NimbRo-OP2X we increased the available space in the trunk to fit a standard Mini-ITX motherboard, with a

¹ RoboCup 2019 NimbRo highlights video: <https://youtu.be/ITe-seb4PsA>

² NimbRo-OP2X hardware: <https://github.com/NimbRo/nimbro-op2>

³ NimbRo-OP2X software: https://github.com/AIS-Bonn/humanoid_op_ros

i7-8700T CPU and a GTX 1050 Ti GPU. Other significant differences With the NimbRo-OP2X, a new type of Dynamixel X actuator—the XH540—was used. Equipped with thicker gears and a fully metal casing, the servomotors are more durable and reliable, compared to the MX-106 from the NimbRo-OP2. Both robots use external gearing to produce the required torque in the leg roll and yaw joints. Initially, they were custom-milled out of brass for the OP2. Due to weight and procurement factors, the OP2X utilizes 3D-printed double-helical gears, which are fast to manufacture.

Copedo. Copedo was built using milled carbon composite and aluminum parts. These provide the necessary rigidity, while keeping the total weight down. Initially, Copedo was built (in 2012) as a TeenSize robot. With the introduction of one vs. one games in the AdultSize class in 2017, we have rebuilt him to have a weight of 10.1 kg and a height of 131 cm [4]. Dynamixel EX-106+ were chosen to power the 5 DoF legs. The legs are additionally equipped with tension springs, which allow for energy storage during locomotion. The 1 DoF arms and 2 DoF neck use RX-64 servos, due to lower torque and speed requirements for their joints. A small, light-weight and efficient Intel NUC NUC7I7BNH (i7-7567U CPU) computer was fitted into Copedo to complete the build.

3 Deep Learning Visual Perception

Our visual perception pipeline improved significantly since RoboCup 2018. Thanks to our new unified perception convolutional neural network (NimbRoNet2), we now can reliably perceive the environment in extremely low and very bright lighting condition. The visual perception system can recognize soccer-related objects, including a soccer ball, field boundaries, robots, line segments, and goalposts through the usage of texture, shape, brightness, and color information.

Our deep-learning-based visual perception system is robust against brightness, viewing angles, and lens distortions. To achieve this, we designed a unified deep convolutional neural network to perform object detection and pixel-wise classification with one forward pass. After post-processing, we managed to outperform our previous non-deep learning approach to soccer vision [5] as well as our previous deep-learning-based model [2]. Our perception system is also able to track [6] and identify our robots [7].

The system has two output heads; one for object detection, and the other for pixel-wise segmentation. The detection head gives the location of the ball, robots, and goalposts. The segmentation head is for line and field detection. Our model uses an encoder-decoder architecture similar to pixel-wise segmentation models like SegNet [8], and U-Net [9]. Due to computational limitations and the necessity of real-time perception, we have made several adaptations, e.g., using a shorter decoder than the encoder. Thus, the number of parameters has been reduced for the cost of losing fine-grained spatial information which can be alleviated using sub-pixel post-processing. To minimize annotation efforts, we utilized transfer-learning. A pre-trained ResNet-18 is chosen as the encoder. Since ResNet was

originally designed for recognition tasks, we removed the Global Average Pooling (GAP) and the fully connected layers in the model. Transpose-convolutional layers are used for up-sampling the representations. To use location-dependent features, we used newly proposed location-dependent convolutional layer [10]. In order to limit the number of parameters used, a shared learnable bias between both output heads is implemented. The proposed visual perception architecture is illustrated in Fig. 2.

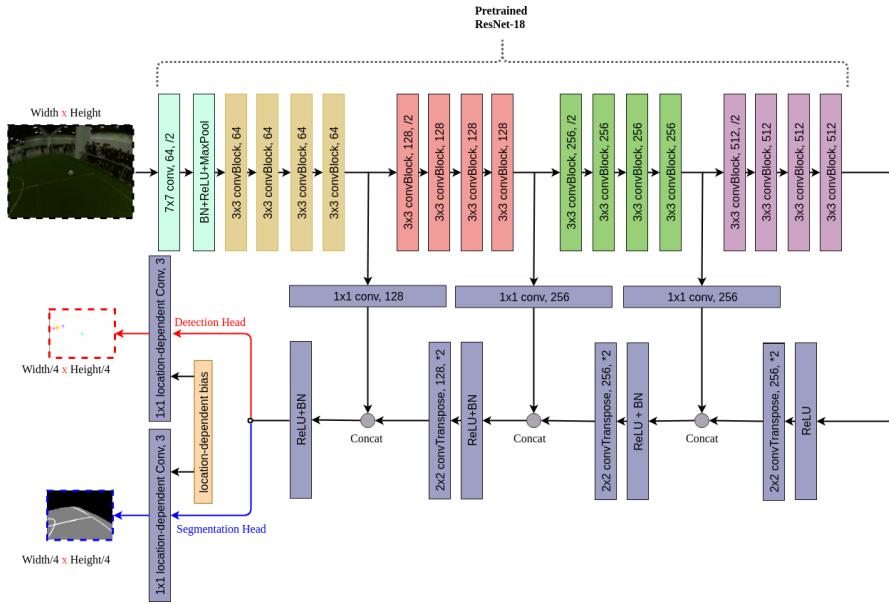


Fig. 2. NimbRoNet2 architecture. Similar to ResNet, each convBlock consists of two convolutional layers followed by batch-norm and ReLU activations. For simplicity, residual connections in ResNet are not depicted. Note that instead of a convolutional layer we used a location-dependent convolution in the last layer.

Different losses were used for different network heads. For detection head, similar to SweatyNet [11], the mean squared error is employed. The target is constructed by Gaussian blobs around the ball center and bottom-middle points of the goalposts and robots. In contrast to last year model, NimbRoNet2 uses a bigger radius for robots with the intuition that annotating a canonical center point is more difficult, thus a bigger radius would less penalize the network for not outputting the exact human labels. In the classification head, we used pixel-wise Negative Log Likelihood. We also added Total Variation loss to the output of all result channels except the line segmentation channel. Total Variation loss encouraged blob response thus helped to have less false positives, especially in field detection.

One other difficulty of this year of RoboCup was very thin goalposts which were hard to detect. However, with many training samples, the network finally

managed to learn it very robustly. After sufficient training, goal posts were detected even when they were hard to recognize by a human. This might be explained by inferring their presence from other features of the pitch like field boundary and lines. One detected hard-to-recognize goal post is shown in the last row of Fig. 3.

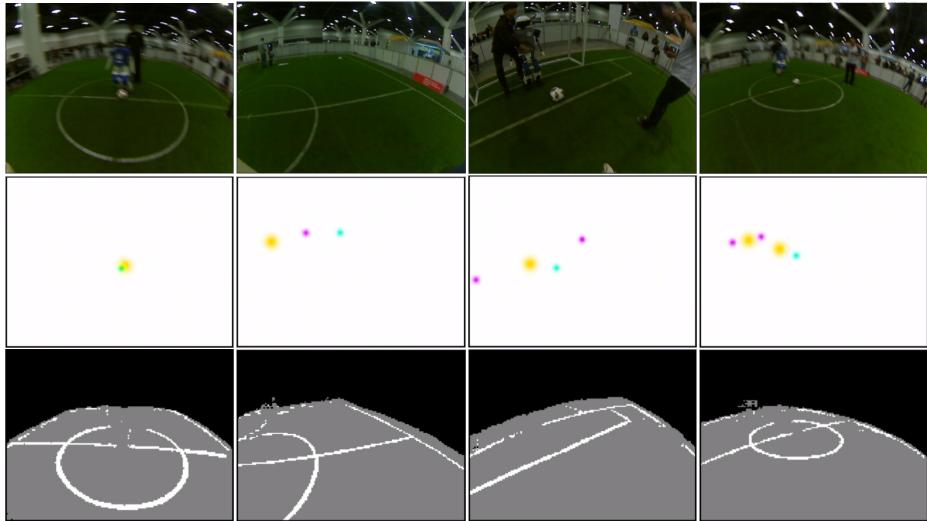


Fig. 3. Object detection results. Upper row: captured images by our robots. Middle row: the output of the network with balls (cyan), goal posts (magenta), and robots (yellow). Bottom row: the output of the segmentation branch with lines (white), field (gray), and background (black).

Despite using Adam optimizer, which has an adaptable per-parameter learning rate, finding a suitable learning rate is still a challenging prerequisite for training. To determine an optimal learning rate, we followed the approach presented by Smith et. al. [12]. Each batch contained only some samples for one of the output heads. We used progressive image resizing that uses small pictures at the beginning of training, and step by step increase the dimensions as training progresses, a method inspired by Brock et. al. [13] and by Yosinski et. al. [14]. In early iterations, the inaccurate randomly initialized model will make fast progress by learning from large batches of small pictures. Within the initial fifty epochs, we used downsampled training images, whereas the weights on the encoder part are frozen. Throughout the following fifty epochs, all parts of the models are jointly trained. In the last fifty epochs, full-sized pictures are used to learn fine-grained details. A lower learning rate is employed for the encoder part, with the intuition that the pre-trained model needs less training time to converge. With the described method, the entire training process with around 9k samples takes less than three hours on a single Titan X GPU with 12 GB of memory. Examples from the test set are pictured in Fig. 3. To annotate more

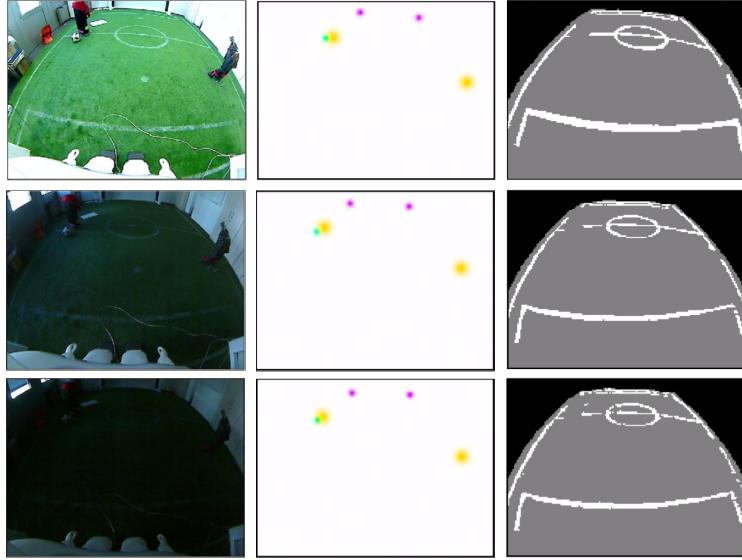


Fig. 4. Object detection under various lighting conditions. Left column: captured images by our robots. Middle column: output of the network indicating balls (cyan), goal posts (magenta), and robots (yellow). Right column: output of the segmentation branch showing lines (white), field (gray), and background (black).

data as quickly as possible, we designed an annotation tool which automatically annotates the input based on the previously trained model. The user then only had to correct those samples which were wrongly classified. This semi-automatic annotation tool was crucial for us to gather as many samples as possible from the RoboCup 2019 environment.

The output of the network is of lower resolution and has less spatial information than the input image. To account for this effect in the detection part, we calculate sub-pixel level coordinates based on the center of mass of a detected contour. There was no need to account for lower resolution output in the field and line segmentation.

After detecting soccer-related objects, we filter them and project each object location into egocentric world coordinates. Using NimbRoNet2, we can detect objects which are up to 10 meters away. The complete perception pipeline, including a forward-pass of the network, takes approximately 36 ms on the robot hardware. Using a unified network helped both detection and segmentation. The network learned to exclude the balls which were outside of the field, hence reducing false detection rate. Outside field object removal was previously done only after post-processing. In addition, the robot was able to play soccer in pitch black, and the perception was robust in different lighting conditions, including direct sunlight and without ambient light (Fig. 4). Unfortunately, this year, all AdultSize games were played with artificial light, thus we could not test our new development for lighting conditions during the competition.

Table 1. Results of the detection branch of our visual perception network.

Type	F1	Accuracy	Recall	Precision	FDR
Ball (NimbRoNet2)	0.998	0.996	0.996	1.0	0.0
Ball (NimbRoNet)	0.997	0.994	1.0	0.994	0.005
Ball (SweatyNet-1 [11])	0.985	0.973	0.988	0.983	0.016
Goal (NimbRoNet2)	0.981	0.971	0.973	0.988	0.011
Goal (NimbRoNet)	0.977	0.967	0.988	0.966	0.033
Goal (SweatyNet-1 [11])	0.963	0.946	0.966	0.960	0.039
Robot (NimbRoNet2)	0.979	0.973	0.963	0.995	0.004
Robot (NimbRoNet)	0.974	0.971	0.957	0.992	0.007
Robot (SweatyNet-1 [11])	0.940	0.932	0.957	0.924	0.075
Total (NimbRoNet2)	0.986	0.986	0.977	0.994	0.005
Total (NimbRoNet)	0.983	0.977	0.982	0.984	0.015
Total (SweatyNet-1 [11])	0.963	0.950	0.970	0.956	0.043

Table 2. Results of the semantic segmentation of our visual perception network.

Type	Accuracy	IOU
Field	0.986	0.975
Lines	0.881	0.784
Background	0.993	0.981
Total	0.953	0.913

Our visual perception pipeline is compared on different soccer-related objects against SweatyNet [11] and our previous model NimbRoNet [2] (Table. 1). We also evaluated our segmentation head (Table. 2). We have outperformed SweatyNet and NimbRoNet, whose results were one of the best-reported in terms of detecting soccer objects. This achievement was also accompanied by being approximately two times faster than SweatyNet in training phase. The reduced training time can be attributed to the progressive image resizing and transfer learning techniques.

4 Robust Omnidirectional Gait with In-walk Kick

Team NimbRo has developed a motion and a gait control framework capable of absorbing pushes from any direction at any time during the gait cycle. This year, for the first time, our NimbRo-OP2(X) adult-sized platforms incorporated step-based push recovery capabilities and in-walk kicks.

Compliant Actuation. Motions performed by the robot are sensitive to the tracking capabilities of the control system. We developed a feed-forward control scheme which modifies the joint trajectories based on the commanded position

and inverse dynamics [15]. The model incorporates factors such as battery voltage, joint frictions, and body inertias.

Open-loop Walking. The walking gait is based on an open-loop central pattern generator calculated from a gait phase angle proportional to the desired gait frequency. To formulate this open-loop gait, we use three different spaces: joint space, Cartesian space and abstract space [16]. The open-loop gait is further extended by the integration of an explicit double support phase, modification of the leg extension profiles, and velocity and acceleration-based leaning strategies [17]. These extensions resulted in passive damping of oscillations and smooth transition between swing and support phases.

Feedback Mechanisms. Several basic feedback mechanisms, namely arm angle, hip angle, continuous foot angle, support foot angle, CoM shifting, and virtual slope, have been built around the open-loop gait core to stabilize the walking [17]. These PID-like feedback mechanisms derive from the state estimation and add corrective action components to the central pattern generated waveforms.

Capture Steps Gait. We use the Capture Step Framework [18,19] to make our robots recover balance. The Capture Step Framework is a composition of central pattern-generated open-loop step motions and a linear inverted pendulum model-based balance controller. In each iteration of the motion control loop, timing and location of the next footstep are computed using the linearized equations of the inverted pendulum model such that the Center of Mass (CoM) would return to a stable limit cycle while also following a commanded walking velocity. Our robots showed stable walking throughout the competition, including balance-restoring capture steps after collisions in games and excelled in the technical challenge.

4.1 In-walk Kick

Our in-walk kick approach integrates the kick directly into the gait to avoid unnecessary stops (Fig. 5), which were required in our previous kicking motions.



Fig. 5. In-walk kick performed during a soccer match.

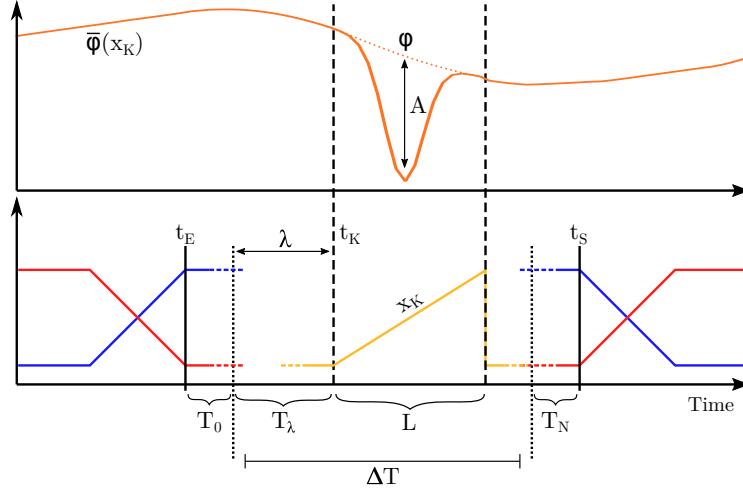


Fig. 6. A schematic visualization of the kick phase x_K and the sagittal leg angle $\bar{\varphi}$. The upper part shows the augmented leg angle $\bar{\varphi}$ in sagittal direction during a kick, where the vertical dashed lines symbolize the start and the end of the kicking motion. The lower part displays the timing parameters used to calculate x_K , where the red line resembles the support coefficient of the kicking leg and the blue line of the supporting leg respectively. The timing parameter λ can be used to move the motion of length L inside the legal execution window of size ΔT .

Thus, a significantly boosting of the overall pace of the game on the larger field has been achieved.

A general schematic of the approach can be seen in Fig. 6. In our approach, an allowed time window ΔT is defined for the kick. Generally, a kick can be performed between the end t_E of the previous support transition phase and the start t_S of the next transition phase respectively. Nevertheless, for $T_0, T_N > 0$, it is advantageous to prohibit kicks in boundary intervals $[t_s, t_s + T_0]$ and $[t_E - T_N, t_E]$ to prevent unwanted foot contact with the ground during the kick execution. In this manner, we define

$$\Delta T = t_E - t_S - T_0 - T_N \quad (1)$$

as the length of the interval where we can safely perform a kick. In addition, given a motion duration $L < \Delta T$, it is possible to perform the kick inside an arbitrary location of the allowed interval. This enables us to define a timing parameter $0 \leq \lambda \leq 1$ resulting in a delay interval

$$T_\lambda = \lambda (\Delta T - L), \quad (2)$$

controlling the starting time of motion execution inside the allowed time window. This is particularly important for the soccer behaviors, since it allows for exact control of the location of the foot at the start and apex of the kick. Altogether,

this results in the actual starting time of the kick

$$t_K = t_S + T_0 + T_\lambda. \quad (3)$$

Consequently, a kick phase x_K is defined, which linearly interpolates from -1 to 1 between the nominal start and end of the kicking motion:

$$x_K(t) = 2 \frac{t - (t_S + T_0 + T_\lambda)}{L} - 1. \quad (4)$$

In the end, the kick phase is used to compute the augmented sagittal leg angle:

$$\bar{\varphi}(t) = \begin{cases} \varphi - A \exp\left(-\frac{1}{2}\left(\frac{x_K(t)}{\sigma}\right)^2\right), & \text{if } t \in [t_K, t_K + L] \\ \varphi, & \text{otherwise} \end{cases} \quad (5)$$

by subtracting a Gaussian curve of the sagittal leg motion, where A defines the amplitude and σ controls the width of the Gaussian, achieving a smooth but distinct forward motion of the leg. The Gaussian term is set to zero beyond the interval boundaries $[t_K, t_K + L]$. Thus, σ has to be small enough such that the activation of the Gaussian can be neglected at the borders, ensuring that the transition to the kick is smooth.

5 Soccer Behaviors

We refer to soccer behaviors to the decision process required to play football. These decisions include, for example, to search for the ball if this is not detected, to go for the ball if we are far from it, or to activate the kick if all the conditions for kicking are granted. The decision process is modeled as a hierarchical Finite State Machine (FSM) with two main layers [20]. The state of the upper layer is established by the state of the game defined in part by the game controller and the role of the player (goalie, striker or defender). The states of the lower FSM represent individual skills of the players such as: move, stop, kick, dribble, dive, among others. Collision avoidance, i.e., avoidance of other robots—either from the opponent team or our team—is part of the lower state machine.

Team Play Strategies. In general, the function of the team play is to safely assign the game roles to each of the players. In this manner, for example, having two strikers simultaneously is not desired in order to avoid collisions between robots of the same team when they are going for the ball. The task assignment is implemented as a server/client architecture where the striker is the server and it is the only one allowed to accept task renegotiation requests. The other players, i.e., defenders and goalies, are allowed to make requests if they find themselves in a better position than the striker, e.g., being closer to the ball. During drop-in games, no task renegotiation was allowed, mainly due to the lack of game roles of other teams. Thus, our players were assigned to fixed roles from the beginning of the match. However, team capabilities were still exhibited during drop-in games, e.g., by our goalkeepers clearing out balls and returning to their corresponding goal. For a deeper discussion about the team play strategies, please refer to [20].

6 Technical Challenges

Technical challenges is a separate competition, where robots have to perform isolated independent tasks during a limited time period. Since the time period for executing all tasks is limited to only 25 minutes, robustness and reliability have the highest importance when designing a solution for each challenge. At RoboCup 2019 there were four technical challenges: push recovery, high jump, high kick and goal kick from moving ball.

6.1 Push Recovery

In this *Push Recovery* challenge, a robot has to withstand three pushes in a row while walking on spot. The pushes are performed by releasing a previously retracted pendulum which then hits the robot at the height of the CoM. The pushes are performed randomly from the front and from the back. The weight of the pendulum is 5 kg and the robots are ranked by the distance of pendulum retraction for the series of three successful attempts. The Capture Step Framework allowed our robot to withstand very strong pushes, making a series of capture steps to regain balance (Fig. 7), and to finish first in this challenge.

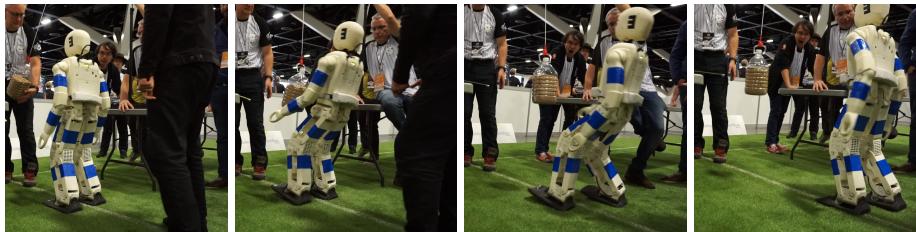


Fig. 7. Technical Challenge: Push Recovery. Several capture steps allow the robot to regain balance after a very strong push of a 5 kg pendulum.

6.2 High Jump

The goal of the *High Jump* is to remain airborne during a vertical jump as long as possible and upon landing remain in a stable sitting or standing posture. For this challenge, motions were pre-designed using key-frames and a simple geometry-based mass distribution principle [21]. By lowering and rapidly lifting the CoM, the accumulated linear momentum at full CoM height propels the robot into the air. After the leap is performed, there is a possibility to decrease the CoM height, which results in folding the legs. This would increase the time in the air by postponing the contact with the ground plane even with a weaker jump upwards. However, we observed that this rapid leg movement caused the robot to often lose balance upon landing. Due to the bent knees, the force upon impact was also damaging to the gears in the knee actuators. In our experience, we have

found that landing on extended legs increased the durability of the actuators and made the landing more reliable. This was largely due to the integrated tension springs in the legs. They provide passive compliance during landing and also contributed greatly to the strength of the jump. Our robot remained airborne for 0.262s and came in second with a difference of 13 ms to the first.

6.3 High Kick

In the *High Kick* challenge, the robot has to score a goal over an obstacle which is positioned on the goal line. The ball is initially positioned at the penalty mark. The goal is only valid if the ball surpassed the obstacle without touching it. The height of the obstacle can be adjusted and the teams are ranked by the height of the successfully over-kicked obstacle. Since it is allowed to touch the ball multiple times, we first move the ball close to the goal line by executing a pre-designed kicking motion. Having the ball close to the obstacle, we execute a pre-designed high kick motion. During this motion the tip of the foot makes first contact with the ball as close to the ground plane as possible. From that point, the foot moves forward and upwards. In order to improve the efficiency of this motion, we use a modified foot with a "scoop" shape. It ensures a prolonged contact with the ball during the high kick motion and—hence—transfer of more energy to the ball, which allows to kick over higher obstacles. Our team came in second in this challenge, successfully kicking over an obstacle of 26 cm height.

6.4 Goal Kick from Moving Ball

The goal of this challenge is to score a goal by kicking a moving ball. The robot is placed at the penalty mark. The ball is positioned at the corner of the field and is passed to the robot either by a human or by another robot. The teams are ranked by the number of successful goals out of three consecutive attempts. In order to know when the robot has to kick, we predicted the time-of-travel of the ball to get in front of the robot foot by estimating its velocity and acceleration from a series of consecutive ball detections, separated in time by a time interval $\epsilon = 0.1$ s. Our robot performing this challenge is shown in Fig. 8. Our team took the first place in this challenge, successfully scoring the goal from the moving ball three out of three times.



Fig. 8. Technical Challenge: Goal Kick from Moving Ball.

7 Game Performance

During the AdultSize 2 vs. 2 Soccer competition of RoboCup 2019, our robots scored 48 goals while receiving none. The robots have shown outstanding performance during the whole tournament including winning the final game 8:0. While 2 vs. 2 competition games have shown individual and team capabilities, drop-in games demonstrated individual skills of each single robot. In the Drop-in tournament, our robots scored 31:7 goals in 6 drop-in games—resulting in winning 57 points with a margin of 33 points to the second best team. Compared to the soccer tournament, we received goals during drop-in games mainly due to the lack of a second field player (our partner teams normally placed goalkeepers), and due to the lack of diving motions when our robots were goalkeepers. The capabilities of our robots were once more demonstrated by winning the AdultSize Technical Challenges. Consequently, NimbRo received the 2019 Best Humanoid Award of the Humanoid League.

8 Conclusions

In this paper, we presented the approaches that lead us to win all possible competitions in the AdultSize class for the RoboCup 2019 Humanoids League in Sydney: soccer tournament, drop-in games and technical challenges. Special emphasis was put on the deep learning based computer vision system that lead our robots to be robust against different lighting conditions and to detect reliably balls up to 10m. Part of our success in the games was explained by the novel in-walk kick, making our games very dynamic and hard to counteract by the opponents. We also presented a step-based push recovery approach that was demonstrated during the competition with impressive performance. Finally, the decision making process and team play strategies were presented, which are responsible for integrating and making use of all individual components.

Acknowledgements

This work was partially funded by grant BE 2556/13 of German Research Foundation.

References

1. G. Ficht, P. Allgeuer, H. Farazi, and S. Behnke, “NimbRo-OP2: Grown-up 3D printed open humanoid platform for research,” in *17th IEEE-RAS Int. Conference on Humanoid Robots (Humanoids)*, 2017.
2. G. Ficht, H. Farazi, A. Brandenburger, D. Rodriguez, D. Pavlichenko, P. Allgeuer, M. Hosseini, and S. Behnke, “NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot,” in *18th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2018.
3. H. Farazi, G. Ficht, P. Allgeuer, D. Pavlichenko, D. Rodriguez, A. Brandenburger, M. Hosseini, and S. Behnke, “NimbRo Robots Winning RoboCup 2018 Humanoid AdultSize Soccer Competitions,” *RoboCup 2018: Robot World Cup XXII, Lecture Notes in Computer Science*, 2019.

4. G. Ficht, D. Pavlichenko, P. Allgeuer, H. Farazi, D. Rodriguez, A. Brandenburger, J. Kuersch, and S. Behnke, “Grown-up NimbRo Robots Winning RoboCup 2017 Humanoid AdultSize Soccer Competitions,” pp. 448–460, 2018.
5. H. Farazi, P. Allgeuer, and S. Behnke, “A monocular vision system for playing soccer in low color information environments,” in *10th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots*, 2015.
6. H. Farazi and S. Behnke, “Real-time visual tracking and identification for a team of homogeneous humanoid robots,” in *RoboCup 2016: Robot World Cup XX*. Springer, 2016, pp. 230–242.
7. ——, “Online visual robot tracking and identification using deep LSTM networks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
8. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2015.
9. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015.
10. Niloofar Azizi, Hafez Farazi and S. Behnke, “Location dependency in video prediction,” in *International Conference on Artificial Neural Networks (ICANN)*, 2018.
11. F. Schnakenburger, M. Scharffenberg, M. Wüller, U. Hochberg, and K. Dorer, “Detection and localization of features on a soccer field with feedforward fully convolutional neural networks (FCNN) for the Adult-size humanoid robot Sweaty,” in *Proceedings of the 12th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2017.
12. L. N. Smith, “Cyclical learning rates for training neural networks,” in *Applications of Computer Vision (WACV), IEEE Winter Conference on*, 2017, pp. 464–472.
13. A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Freezeout: Accelerate training by progressively freezing layers,” *arXiv:1706.04983*, 2017.
14. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3320–3328.
15. M. Schwarz and S. Behnke, “Compliant robot behavior using servo actuator models identified by iterative learning control,” in *RoboCup 2013: Robot World Cup XVII*, 2013.
16. S. Behnke, “Online trajectory generation for omnidirectional biped walking,” in *Proceedings of 2006 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.
17. P. Allgeuer and S. Behnke, “Omnidirectional bipedal walking with direct fused angle feedback mechanisms,” in *16th IEEE-RAS Int. Conference on Humanoid Robots (Humanoids)*, 2016.
18. M. Missura, “Analytic and learned footstep control for robust bipedal walking,” Ph.D. dissertation, Universitäts-und Landesbibliothek Bonn, 2016.
19. M. Missura and S. Behnke, “Walking with capture steps,” in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 526–526.
20. D. Rodriguez, H. Farazi, P. Allgeuer, D. Pavlichenko, G. Ficht, A. Brandenburger, J. Kuersch, and S. Behnke, “Advanced soccer skills and team play of RoboCup 2017 TeenSize winner NimbRo,” in *RoboCup 2017: Robot World Cup XXI*. Springer, 2017, pp. 435–447.
21. G. Ficht and S. Behnke, “Online Balanced Motion Generation for Humanoid Robots,” in *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018.