Stability Controller on the Atlas Robot Example

Mikołaj Wasielica
Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland

Abstract: The paper presents the gait framework for a biped robot on the Atlas robot example. The method utilizes inverted pendulum model and static stability controller with correction from IMU sensor. A straight-forward balance control strategy based on ankle joints control is proposed. The controller which stabilizes the robot during execution of the planned path is described. To show the efficiency of the proposed method the results obtained in the Virtual Robotics Challenge environment (Gazebo) are provided.

Keywords: Bipeds, Humanoid Robots, Legged Robots, Motion Planning

1. Introduction

In October 2012 Defense Advanced Research Projects Agency (DARPA) announced Robotics Challenge (DRC). Using semi-autonomous robots, teams must solve a series of complex and heterogeneous tasks, such as driving a car and navigating on foot through a rubble pile. The virtual competition, the Virtual Robotics Challenge (VRC) [11], was the first of its kind where teams from around the world would use cloud-hosted resources to complete simulated tasks that mimic disaster response scenarios. Twenty-six teams were selected to participate in the VRC, held in June of 2013. Teams were required to complete 15 tasks grouped into three categories: locomotion, driving and manipulation. The purpose of the VRC was to accelerate the software development using a simulated environment, as well as to reduce the risk of working with complex and expensive robots.

The contest was inspired by the 2011 Fukushima Daiichi nuclear disaster, and carries a USD 2 million prize for the winning team. The inability to send humans into these hazardous areas challenged the robotics research community who failed with operating commercially available robots in human oriented tasks.

A fundamental ability of the biped robot is walking, though in this research we focus on the locomotion of the Atlas robot using VRC framework. We show how using simple methods we are able to generate gait of the complex robot. The proposed methods were experimentally validated in simulation environment.

Autor korespondujący:
Mikołaj Wasielica, mikolajwasielica@gmail.com

Artykuł recenzowany
nadesłany 30.03.2017 r., przyjęty do druku 01.06.2017 r.

1.1. Related Work

Teams taking part in VRC proposed different strategies of motion control. Kohlbrecher et al. [15] had developed open-loop key frame-based quadrupedal locomotion which was used during the rough terrain task and whenever the robot had fallen. Team ViGIRs approach was satisfying on rough terrain but had many limitations in narrow spaces and it reduced the Atlas workspace. The same team has also developed a complete system for supervised footstep planning including perception, world modeling, 3D planner and operator interface to enable a humanoid robot to perform sequences of steps to traverse uneven terrain [21]. The planner utilizes a black box walking controller without knowledge of its implementation details.

Ghassan et al. [2] used artificial neural networks to learn the robots nonlinear dynamics on the fly using a neuroadaptive control algorithm. The learned nonlinear dynamics were utilized along with a filtered error signal to generate input torques to control the system. Results show that the ability to approximate the robot nonlinear dynamics allows for full-body control without the need of modeling such a complex system. However this method handled the competition it is complex and hard to implement.

Stable walking motion and control for humanoid robots are already well investigated research topics. Works, that addressed the balance control in response to unexpected disturbances from the environment, tackled such tasks as: keeping upright posture on a changing slope [6], keeping balance on unstable ground [1], walking on uneven terrain [13], maintaining balance when an obstacle appears suddenly in front of the robot [14, 16, 22]. The main task of [4]. There exists a few techniques of stabilizing a robot body orientation by shoulder joint motion [18]. Human beings can control the postures themselves for unexpected forces. It is known that there are three basic strategies such as ankle strategy, hip strategy and step strategy [7] for keeping balance against unexpected disturbance force. In the field of humanoid robots, suggestion have been made about possible ways to adopt the ankle and hip strategies, so far [5, 3, 17]. These three major balance strategies are widely used in such tasks as inclined plane walking compensation for a humanoid robot [12, 20, 27].

After introducing the Atlas robot in the next section, we describe motion planning strategies in third section. Experimental results are summarized in fourth section. The paper concludes with fifth section.

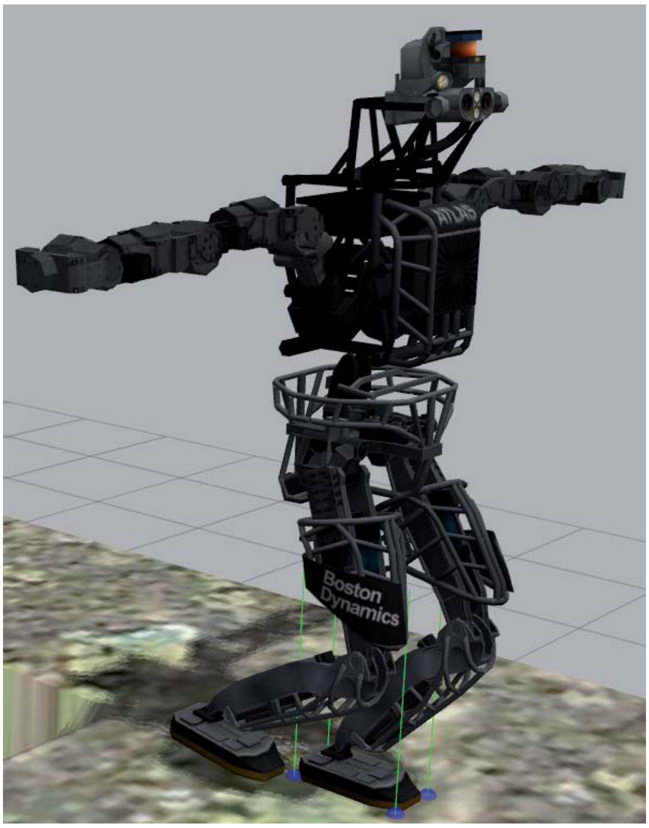

Fig. 1. Boston Dynamics' humanoid robot Atlas in Gazebo
Rys. 1. Robot Atlas firmy Boston Dynamics

2. Atlas Robot

Atlas (Fig. 1) is a full-scale humanoid robot developed by the American robotics company Boston Dynamics for the DRC. The company was funded by Marc Raibert in 1992. Marc Raibert is a former Carnegie Mellon University and MIT professor who developed the first self-balancing hopping robots [19].

The robot is 1.88 m high and weighs about 150 kg. It has 28 hydraulically actuated joints: 6 per arm and leg, 3 in torso, and 1 neck joint. Joints are powered by on-board hydraulic pump and 480 V three-phase external power at 15 kW [9]. Joint positions are reported by linear variable differential transformers (LVDTs) mounted on the actuators. Velocities are computed through numerical differentiation of the position signals, while joint forces are estimated using pressure sensors in the actuators.

The robot is equipped with the Multisense SL sensor head developed by Carnegie Robotics. The head consists a Hokuyo UTM-30LX-EW rangefinder and a stereo camera, providing 3D point cloud [8].

2.1. Simulation Platform

We utilized Darpa Robot Challenge simulator (DRCsim) package under a GAZEBO (universal robot simulation with physics engine). The set of these frameworks models all the mechanical and electrical aspects of Atlas, giving an integrated simulation environment which can be used to simulate the execution of tasks in a disaster scenario. Control algorithm of the robot is realized by the proprietary software based on Robot Operating System (ROS) which is an open-source software framework.

We used RViz to obtain all information about the robot state. It is ROS's built-in visualization tool to display the robots full body joint states (Fig. 2).

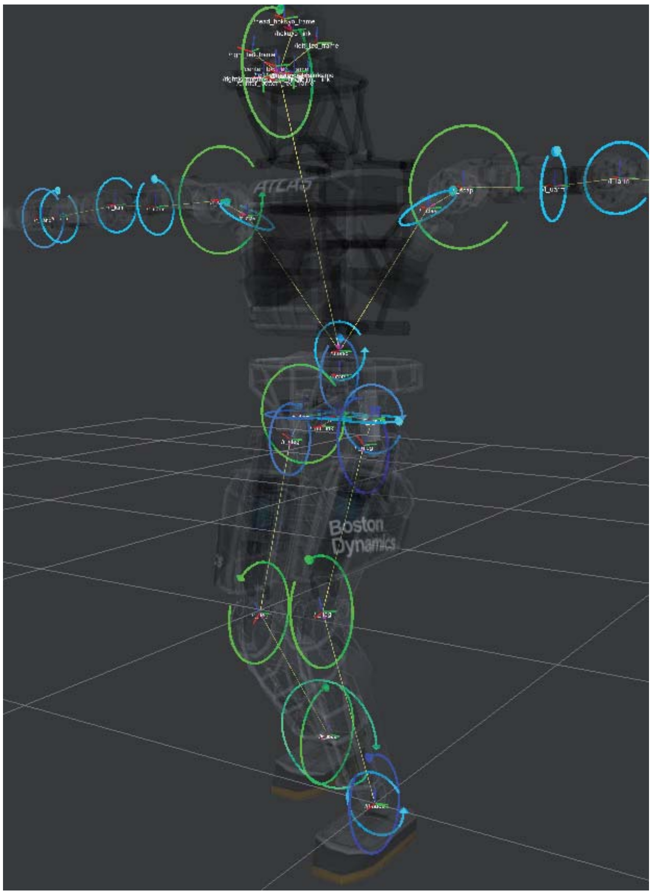

Fig. 2. Sample visualization of DOFs of the Atlas. Torque is represented as circle, its size and color indicates value
Rys. 2. Przykładowa wizualizacja stopni swobody robota Atlas. Momenty sił przedstawione są w postaci okręgów, a ich kolor i wielkość oznacza wartość

3. Motion Planning

3.1. Trajectory Generating

In our task we focused on walking abilities, to be precise and stable. Legs are mainly responsible for the locomotion therefore the robot's upper-body joints (head, arms and trunk) configuration was fixed with arms aside. As well vertical orientation of whole upper-body was constant. Only lower-body parts were involved in gait. To enable motion of the Atlas we measured distances between its joints and derived an algebraic form of Inverse Kinematics (IK) and Forward Kinematics (FK) equations. Dimensions and masses of the robot's parts were obtained from the RViz. Input of the FK is actual configuration of the robot's legs provided by the ROS. Configuration is presented as a set of angular positions of each joint in radians. For example the following equation represents transformation between thigh and calf links located in rotational 1 DoF knee joint:

$$A_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_{12} & -\sin\theta_{12} & 0.422\sin\theta_{12} \\ 0 & \sin\theta_{12} & \cos\theta_{12} & -0.422\cos\theta_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (1)$$

where θ_{12} is configuration of the knee joint.

Combining transformations matrices of the leg's kinematic chain we calculate a result which is 3D Cartesian position and orientation of the foot relatively to coordinate system situated in pelvis of the Atlas. In IK an input is desired position of the

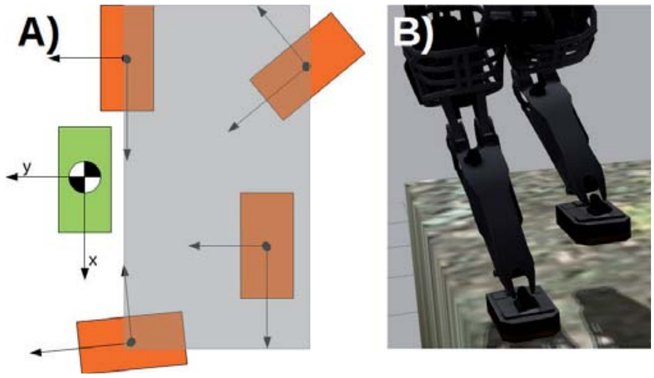

Fig. 3. A) A simplified representation of the kinematic reachability region for footstep placement and several example locations for the right foot (orange) relative to the left foot (green); B) Visualization of reaching a new position by right foot

Rys. 3. A) Uproszczona reprezentacja obszarów dostępnych kinematycznie dla stopy prawej (pomarańczowy) względem stopy lewej (zielony); B) Wizualizacja stawiania prawej stopy w nowej pozycji)

foot and an output adequate leg joints configuration. Both FK and IK equations are derived for each single leg independently, thereby avoiding singular configurations.

To generate smooth motion of the robot we developed two trajectory forming algorithms. First is called *jtraj* and computes a joint space trajectory between two configurations (eq. 2).

$$q = jtraj(q_0, q_1, n) \tag{2}$$

$q(M \times N)$ is a joint space trajectory where the joint coordinates vary from $q_0(1 \times N)$ to $q_1(1 \times N)$. Time is assumed to vary from 0 to 1 in n steps. The trajectory q is $M \times N$ matrix, with one row per time step, and one column per joint. A quintic (5th order) polynomial is used for velocity and acceleration generation. Also trapezoidal velocity profile can be used alternatively.

Second trajectory forming algorithm is called *ctraj* and it computes Cartesian trajectory between two poses (eq. 3).

$$T = ctraj(T_0, T_1, n) \tag{3}$$

$T(4 \times 4 \times N)$ is a Cartesian trajectory from pose T_0 to T_1 with n points. Number of points n is calculated as total distance between poses divided by the adjusted precision. Precision is a minimal distance between two following points on the trajectory, and it is set by the user. Also a quintic (5th order) polynomial is used for velocity and acceleration generation.

Both IK, *jtraj* and *ctraj* functions provide ability to move the robot's foot from any reachable start pose to the end pose. Also it should be noticed that orientation of the feet can be adjusted easily, which is advantage on uneven terrain walk.

3.2. Stabilization Control

To stabilize the Atlas we developed static stability controller, based on its configuration and mass distribution. We focused on the static stability controller, because our aim was to achieve stable gait with high positioning precision. We used model developed in our previous work [24]. We modeled the robot as a single point mass with two force vectors representing the feet. Position of the robot's Center of the Mass (COM) is calculated from the individual body part's masses and its configuration. For each kinematic link we measured weight and relative position of the COM. Using FK we derived equation of total COM position, relatively to Atlas coordinate system.

Robot will maintain the balance till its COM will be inside the support polygon. To achieve this goal we provided ankle strategy to control position of the COM. In the strategy ankles joints configurations are used to orientate the robot's body in single and double-support phase. In other words in single-support phase normal vector to the supporting foot has to point the COM to provide the stability. Orientation of the ankle joints is controlled by a PID controller using IK and *ctraj*.

To improve stability we added correction from Inertial Measurement Unit (IMU). Based on inclination of the robot the algorithm increases or decreases orientation of the individual foot to maintain the COM in the center of the supporting polygon of the feet. Algorithm helps to suppress wobble of the robot.

In our previous work [24] we presented detailed description about our stabilization controller.

3.3. Path Planning

Our goal was to develop motion planner which performs the walk based on the given step locations. Positions of the points are provided in global coordinates related to the map. In our article [23] we present method of computing a sequence of kinematically-reachable and safe step locations using Rapidly-exploring Random Tree (RRT) algorithm, in this work we obtain them manually to have predictable and repeatable paths. Data are saved in *.mdat* file, where each line contains position in meters and orientation as quaternion of the left and right foot as follows: $x_{lf}, y_{lf}, z_{lf}, qw_{lf}, qx_{lf}, qy_{lf}, qz_{lf}, x_{rf}, y_{rf}, z_{rf}, qw_{rf}, qx_{rf}, qy_{rf}, qz_{rf}$.

Using the tool implemented in ROS we were able to obtain actual robot's pelvis pose: 3D position in meters and orientation as quaternion in the map coordinates. To know actual pose of robot's feet we transformed the pelvis pose using FK. Before executing sequent step our algorithm converts the global step location into local coordinate system related to actual supporting foot. In other words, when Atlas stands on the left leg its coordinate origin is situated in the center of the left foot (Fig. 3.A).

Locating origin of Coordinate System (SC) in the center of supporting foot has major advantage that COM of the Atlas is located over this point in single-support phase. Using this knowledge we experimentally examined kinematic reachability region for footstep placement, which actually was a workspace of the leg including stability of the robot. We offline computed size and position of the gray rectangle presented in Fig. 3.A. Before performing each step algorithm checks if the swinging foot desired position is located in the rectangle. This guarantees that the step is executable. If position of foot is outside the safe reachable region algorithm transforms coordinates to new one situated as close as possible to the desired position but in the safe area. The difference in the position will be corrected in the next step to compensate the shift. Figure 3.B presents the situation before performing the step. In this moment our algorithm compares the desired position of the right foot from the file with the actual global position of the robot and adjusts the length and orientation of the step to reach desired position. Therefore we are able to minimize the drift of the robot's global position.

Also orientation of the Atlas obtained from the onboard IMU is used to improve footstep placement, especially in the single--support phase when the Atlas is susceptible to torsion. Without correction Atlas was not able to place the swinging foot on the same height as supporting foot, is the result robot was falling down. Our algorithm takes into account actual orientation of the Atlas and corrects (by rotation) position of the footstep. Because the IMU data are noisy we filter them by a digital filter.

The walking controller runs based on Finite State Machine (FSM) with four cyclic states plus begin state. A walking cycle repeats for the left and right leg, and each cycle consists of two states: lifting and lowering the leg.

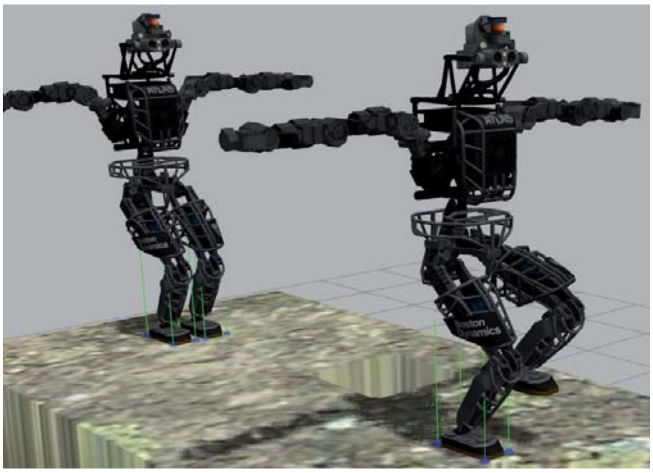

Fig. 4. Straight walk (green lines indicate foot pressure to the ground)
Rys. 4. Chód na wprost (linie przedstawiają siłę nacisku na podłoże)

4. Experimental Results

To examine the proposed control framework, two experiments were designed to validate the effectiveness. Our algorithms were tested on the simulated humanoid robot Atlas in the dynamics simulator Gazebo. Atlas is equipped with IMU as well as stereoscopic cameras and LIDAR, but we only use the IMU data. The robot has 28 actuated joints with position and angular velocity information available at each joint, but the upper-body joints have constant configuration during experiment. The simulation runs on a standard PC computer with Intel i7 processor, 4 GB RAM memory and AMD Radeon graphics card. The computer runs Linux Ubuntu 12.04 operating system.

4.1. Walking with Constant Orientation

First, the walking experiment with the constant orientation of the robot was performed. The robot was not able to turn, so it could move only forward and backward, to the right and to the left. The robot was placed on the start position and had to walk to the end of the path (Fig. 4). The path included 3D global position of each foot described as following steps from the beginning, saved in the text file form. The position of single points was calculated manually off-line having regard their enforceability. The ground was flat and leveled.

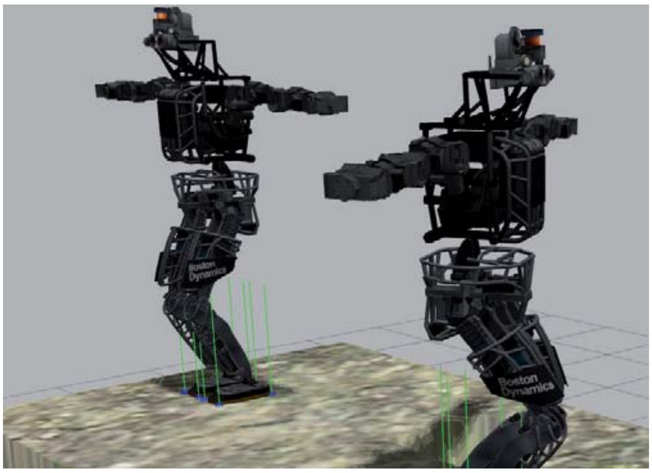

Fig. 6. Walking with turning (green lines indicate foot pressure to the ground)
Rys. 6. Chód po łuku (zielone linie przedstawiają siłę nacisku na podłoże)

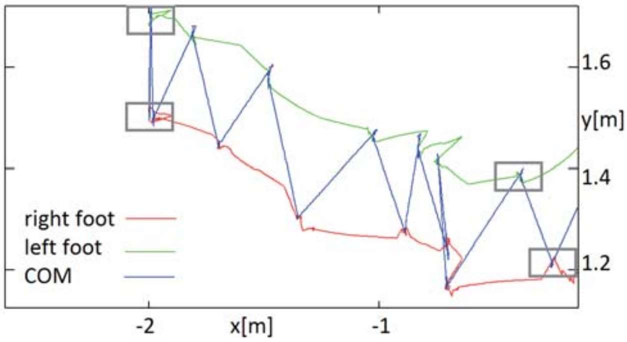

Fig. 5. Trajectory of straight walk (top view)
Rys. 5. Trajektoria chodu na wprost (widok z góry)

Figure 5 presents collected data from the first experiment. On the left, initial position of the feet is marked. Green and red lines indicate trajectory of each foot respectively, blue line is the position of COM of the robot. Coordinates are presented from the top view.

Atlas performed the first two steps in place to adjust its initial position and then started to walk ahead putting its feet alternately. We noticed sliding motion during performing the single leg stand. It is visible on the Fig. 5 as drift of the foot and COM position. The COM was on the edge of the foot then. This problem was mainly caused by the imperfect simulation environment. To improve stability of the robot we adjusted friction and contact parameters in the configuration files, but it was not possible to eliminate slippering completely. Also in single support phase the robot was swaying, what was suppressed by our stabilization control algorithm based on IMU data. Widely spaced hands helped to improve balance of the robot.

Performing single step took the robot 12 s. This long time is a compromise between stability and performance, because slower the robot walks, more stable it is. Our aim was to complete the path regardless of the time. Thanks to this strategy the Atlas robot falls down only when it places its foot outside or on an edge of the experimental platform.

The robot has not fall down during our experiments even once.

4.2. Walking with Body Rotation

The second experiment aimed to validate robot's turning abilities. The same terrain as in the first experiment was used. Also Atlas was placed in the same start point as in the first experiment, but with different orientation. Its body was rotated by 90° to the right. At the end of the path robot was oriented as presented on the Fig. 6. The path also was generated manually, but now it included additional orientation of each

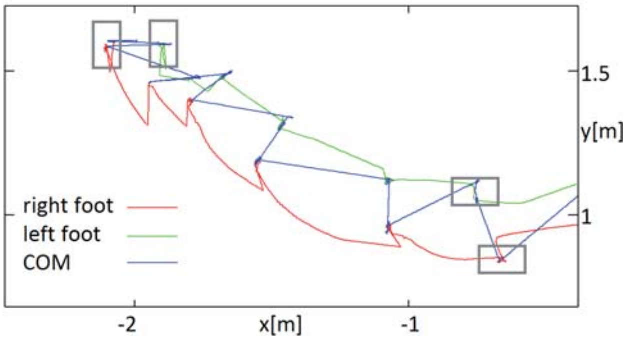

Fig. 7. Trajectory of walk with turning (top view)
Rys. 7. Trajektoria chodu po łuku (widok z góry)

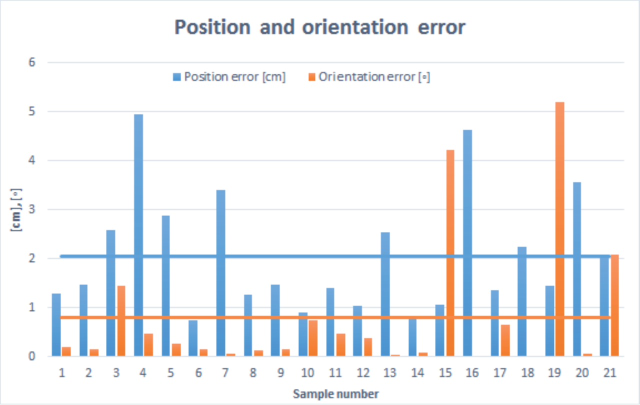

Fig. 8. Position [cm] an orientation [°] error obtained from the experiment, horizontal lines represent an average value
Rys. 8. Błąd pozycji [cm] i orientacji [°] uzyskany w eksperymentach, poziome linie przedstawiają wartości średnie

foot. Orientation was provided as quaternion. The robot started its move (Fig. 7 – on the left) by adjusting initial position and orientation. Then Atlas was moving and rotating its legs at the same time. Additional rotation had no influence to the feet slippering or the body swaying.

4.3. Position and Orientation Error

For each experiment information of reference and real position as well as orientation of the robot's feet was collected. The data of the foot positioning error were collected in single-support phase, while robot was touching the ground and was extremely unstable. The moment that the early contact of the swing leg occurred was the actual beginning of the double-support phase. From this moment, both feet of the robot contacted the ground, which means that the actual stable region was the area bounded by both feet that were touching the ground. Thus, in the double support phase robot was more stable.

Figure 8 presents position and orientation displacement error during the second experiment. Abscissa axis includes number of step while ordinate axis shows positioning error in [cm] and orienting error in [°]. Additional horizontal lines describe averages of samples. Results show that robot's foot was displaced from the desired position maximally 4.96 cm, while average error was 2.05 cm. Maximal error in rotation was equal 5.21° and average was 0.81°. Both deviations are small and satisfying, which confirms reliability of our algorithm.

5. Conclusion and Future Work

In this paper gait control framework for Atlas robot is proposed. It is suggest to rely on the inverted pendulum model representation of the robot. A straight-forward balance control strategy based on ankle joints control is proposed and it proved to be efficient and accurate. This method is not computationally complex and needs no further optimization of configuration. Experimental results demonstrate that implementation successfully controls the Atlas robot so that it follows the given path while maintaining the balance. In future work we plan to validate our methods on real humanoid robot.

References

1. Abdallah M., Goswami A., *A Biomechanically Motivated Two-Phase Strategy for Biped Upright Balance Control*, [in:] Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, 1996–2001, DOI: 10.1109/ROBOT.2005.1570406.
2. Atmeh G.M., Ranatunga I., Popa D.O., Subbarao K., Lewis F., Rowe P., *Implementation of an Adaptive, Model Free, Learning Controller on the Atlas Robot*, [in:] Proceedings of the 2014 American Control Conference (ACC), Portland, USA, 2014, 2887–2892, DOI: 10.1109/ACC.2014.6859431.
3. Azevedo M.C., Poignet P., Espiau B., *Artificial Locomotion Control: From Human to Robots*, "Robotics and Autonomous Systems", Vol. 47, Iss. 4, 2004, 203–223, DOI: 10.1016/j.robot.2004.03.013.
4. Fujiwara K., Kanehiro F., Kajita S., Hirukawa H., *UKEMI: Falling Motion Control to Minimize Damage to Biped Humanoid Robot*, [in:] Proceedings of the 2002 IEEE International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, Oct. 2002, 2521–2526, DOI: 10.1109/IRDS.2002.1041648.
5. Guihard M., Gorce P., *Dynamic Control of Bipeds Using Ankle and Hip Strategies*, [in:] Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002, 2587–2592, DOI: 10.1109/IRDS.2002.1041660.
6. Gutmann J.S., Fukuchi M., Fujita M., *Stair Climbing for Humanoid Robots Using Stereo Vision*", [in:] Proceedings of the 2004 IEEE International Conference on Intelligent Robots and Systems, Sendai, Japan, 2004, 1407–1413, DOI: 10.1109/IROS.2004.1389593.
7. Hofmann A., *Robust Execution of Bipedal Walking Tasks from Biomechanical Principles*, Ph.D. dissertation, Massachusetts Institute of Technology, January 2006.
8. http://carnegierobotics.com/multisense-sl/
9. http://spectrum.ieee.org/automaton/robotics/humanoids/darpaunveils-atlas-drc-robot
10. http://www.theroboticschallenge.org/
11. Defense Advanced Research Projects Agency (DARPA), *DARPA Robotics Challenge (DRC) and Virtual Robotics Challenge (VRC)*, 2015, http://theroboticschallenge.org/about.
12. Kaewlek N., Maneewarn T., *Inclined plane walking compensation for a humanoid robot*, [in:] Proceedings of the 2010 International Conference on Control, Automation and Systems (ICCAS), Gyeonggido, Korea, 2010, 1403–1407, DOI: 10.1109/ICCAS.2010.5670323.
13. Kajita S., Morisawa M., Harada K., Kaneko K., Kanehiro F., Fujiwara K., Hirukawa H., *Biped Walking Pattern Generator allowing Auxiliary ZMP Control*, [in:] Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 2006, 2993–2999, DOI: 10.1109/IROS.2006.282233.
14. Kaneko K., Kanehiro F., Kajita S., Morisawa M., Fujiwara K., Harada K., Hirukawa H., *Motion Suspension System for Humanoids in case of Emergency - Real-time Motion Generation and Judgment to suspend Humanoid*, [in:] Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 2006, 5496–5503, DOI: 10.1109/IROS.2006.282184.
15. Kohlbrecher S., Conner D.C., Romay A., Bacim F., Bowman D.A., von Stryk O., *Overview of Team ViGIR's Approach to the Virtual Robotics Challenge*, [in:] Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linkoping, Sweden, 2014, 1–2.
16. Morisawa M., Kaneko K., Kanehiro F., Kajita S., Fujiwara K., Harada K., Hirukawa H., *Motion Planning of Emergency Stop for Humanoid Robot by State Space Approach*, [in:] Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 2006, 2986–2992, DOI: 10.1109/IROS.2006.282232.

17. Nenchev D.N., Nishio A., *Ankle and Hip Strategies for Balance Recovery of a Biped Subjected to an Impact*, Robotica, Vol. 26, Iss. 5, 2008, 643–653, DOI: 10.1017/S0263574708004268.
18. Ozel S., Eskimez S.E., Erbatur K., *Humanoid Robot Orientation Stabilization by Shoulder Joint Motion During Locomotion*, [in:] Proceedings of the 9th Asian Control Conference (ASCC), Istanbul, Turkey, 2013, 1–6, DOI: 10.1109/ASCC.2013.6606339.
19. Raibert M., *Legged Robots that Balance*, 1986, MIT Press.
20. Seven U., Akbas T., Fidan K.C., Yilmaz M., Erbatur K., *Humanoid robot walking control on inclined planes*, [in:] Proceedings of the 2011 IEEE International Conference on Mechatronics (ICM), Istanbul, Turkey, 2011, 875–880, DOI: 10.1109/ICMECH.2011.5971237.
21. Stumpf A., Kohlbrecher S., Conner D.C., von Stryk O., *Supervised Footstep Planning for Humanoid Robots in Rough Terrain Tasks using a Black Box Walking Controller* [in:] Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots (Humanoids), Madrid, Spain, 2014, 287–294, DOI: 10.1109/HUMANOIDS.2014.7041374.
22. Tanaka T., Takubo. T., Inoue K., Arai T., *Emergent stop for Humanoid Robots*, [in:] Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 2006, 3970–3975, DOI: 10.1109/IROS.2006.281833.
23. Wasielica M., Belter D., *RRT-based motion planner and balance controller for a biped robot*, Advances in Cooperative Robotics, World Scientific, 2016, 404–411, DOI: 10.1142/9789813149137_0048.
24. Wasielica M., Wąsik M., *Active stabilization of a humanoid robot base on inertial measurement unit data*, International Conference on Mechatronics – Mechatronika (ME), Brno, Czech Republic, 2014, 364–369, DOI: 10.1109/MECHATRONIKA.2014.7018285.
25. Wasielica M., Wąsik M., Kasiński A., Skrzypczyński P., *Interactive Programming of a Mechatronic System: A Small Humanoid Robot Example*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Wollongong, Australia, 2013, 459–464, DOI: 10.1109/AIM.2013.6584134.
26. Wasielica M., Wąsik M., Skrzypczyński P., *Design and applications of a miniature anthropomorphic robot*, “Pomiary Automatyka Robotyka”, Vol. 17, No. 2, 2013, 294–299.
27. Yong-Duk K., In-Won P., Jeong-Ki Y., Jong-Hwan K., *Stabilization control for humanoid robot to walk on inclined plane*, [in:] Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots 2008 (Humanoids), Daejeon, Korea, 2008, 28–33, DOI: 10.1109/ICHR.2008.4755927.

Kontroler równowagi na przykładzie robota Atlas

Streszczenie: Artykuł przedstawia system generowania chodu dla robotów dwunożnych na przykładzie robota Atlas. Metoda wykorzystuje model odwróconego wahadła oraz statyczny kontroler stabilności wraz z korekcją z sensora IMU. Zaproponowano prostą metodę utrzymywania równowagi w oparciu o sterowanie ruchami stóp robota. Opisano też kontroler stabilizujący robota podczas pokonywania zaplanowanej ścieżki. Zweryfikowano działanie zaproponowanych metod na robocie Atlas w symulatorze Virtual Robotics Challenge (Gazebo).

Słowa kluczowe: robot dwunożny, robot humanoidalny, robot kroczący, planowanie ruchu

Mikołaj Wasielica, MSc, Eng.
mikolajwasielica@gmail.com
mikolaj.r.wasielica@doctorate.put.poznan.pl

Graduated (2012) from the Poznan University of Technology. He is a PhD student at Faculty of Electrical Engineering of PUT. His current research interests include: anthropomorphic robots motion planning, motion capture techniques, and man-machine interfaces.