

OPTIMIZATION-BASED APPROACH FOR MOTION PLANNING OF A ROBOT WALKING ON ROUGH TERRAIN

Submitted: 20th January 2013; accepted: 4th June 2013

Dominik Belter

DOI: 10.14313/JAMRIS_4-2013/35

Abstract:

The paper presents a motion planning algorithm for a robot walking on rough terrain. The motion-planner is based on the improved RRT (Rapidly Exploring Random Tree)-Connect algorithm. The Particle Swarm Optimization (PSO) algorithm is proposed to solve a posture optimization problem. The steepest descent method is used to determine the position of the robot's feet during the swing phase. The gradient descent method is used for smoothing the final path. The properties of the motion planning algorithm are presented in four cases: motion planning over a bump, concavity, step and rough terrain mockup. The maximal sizes of particular obstacle types traversable by the Messor robot with the new, optimized motion plan are given.

Keywords: walking robot, motion planning, rough terrain locomotion

1. Introduction

Most of the outdoor mobile robots are equipped with tracks or wheels. This kind of locomotion is very efficient on flat terrain. When the terrain is rough the mobility of such robots is limited by the size of obstacles (rocks, logs, etc.). Moreover, the tracked drive destroys the ground especially when the robot is turning round. Walking is more efficient on rough terrain. Thus, to improve the mobility of legged mobile robots on various types of terrain new motion planning methods are needed. Recently, quadrupeds and bipedal humanoid robots are gaining popularity, but during Urban Search and Rescue (USAR) missions multi legged robots are more suitable due to their inherent static stability.

The nature of legged locomotion is discrete – different than wheeled locomotion. Usually, the wheels of a robot should have continuous contact with the ground, whereas each leg of a walking robot has a sequence of swing and stance phases. The feet of a robot have point-like contact with the ground. As a result, the locomotion task includes a sequence of robot's postures and transitions between these postures (interpreted as discrete states).

Motion planning of a robot walking on rough terrain is a complex task. Walking robot can't be represented as a point in an environment with obstacles. While walking on rough terrain the robot should take into account full shape of the terrain. Collision with the ground as well as static stability and workspace of the robot (a space which the feet of the robot can reach)

should be considered. Finally, motion planning for a walking robot is a problem with many constraints, and the solution has to be determined in a multi-dimensional solution space.

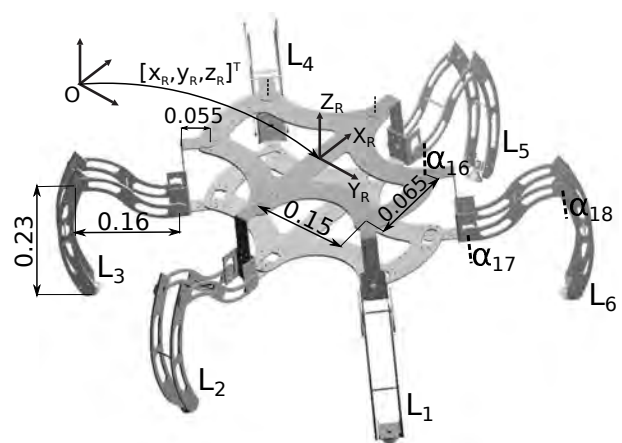


Fig. 1. Kinematic structure of the Messor robot (all dimensions in [m])

The posture of our walking robot (the state of the robot) Messor [24] is determined in a 24-dimensional space. The kinematic structure of the robot is presented in Fig. 1. The position of the robot's body in the reference frame O is determined by a 3D vector $[x_R, y_R, z_R]^T$. Orientation of the robot's body is given by three RPY (roll-pitch-yaw) angles $\theta_R, \varphi_R, \gamma_R$. Each joint position is given by α_i value. The numbering of joints starts from the shoulder servomotor of the first leg, and ends on the joint between tibia and femur of the sixth leg (Fig. 1). The posture is determined by 18 reference values for servomotors located in robot's legs and six values representing position and inclination of the robot's platform ($q = [\alpha_1, \dots, \alpha_{18}, x_R, y_R, z_R, \theta_R, \varphi_R, \gamma_R]^T$). We are looking for methods which can deal with multi-dimensional solution space to find a path for the robot walking on rough terrain. Moreover, the sought algorithm should be fast enough to operate in real time. To find feasible motion of the robot the motion planner should find the way how to go round obstacles and, if necessary, to find the way how to climb obstacles.

Kinematic structure of a robot influences the approach to motion generation. Various approaches are used to obtain efficient locomotion on rough terrain. The rimless wheel structure is used in the RHex robot [21]. The rotational motion of the shaft is converted into walking behavior through flexible spokes. Robust locomotion on rough terrain can be also gen-

erated by using reactive controllers. The example of such a robot is the BigDog [29]. Both methods will fail whenever the robot has to deal with very risky environment e.g. while crossing the stream. Without a foothold selection method the robot will fall down. In our approach we precisely plan a full path of the robot. The robot determines its posture, selects footholds, plans motion of the feet and body. It also checks stability, workspace and avoids collisions with the ground and self-collisions.

By planning full motion of the robot it is also possible to avoid deadlocks (situations when the robot can't move because of the lack of stability or lack of the kinematic margin). Planning motion of the robot also allows to avoid local minima (here understood as a loop of moving backwards and forwards, oscillating around what is in these circumstances a lower-dimensionality local minimum [1]), and to avoid obstacles. This behavior is possible when a reactive controller is used. A reactive controller computes next action taking into account a current context. Thus, the reactive controller might cause 'infinite loops' in robot's behavior, e.g. according to the current context the next motion of the robot is step forward, then step back and again forward, back, etc. To avoid the robot getting trapped in local minima the deliberative paradigm is used in the control software of the machine.

The paper presents recent improvements of the motion planning method for the Messor six-legged robot. This method is based on Rapidly-exploring Random Trees Connect algorithm (RRT-Connect) [16]. During execution of the planned path the robot creates an elevation map of the environment using the Hokuyo URG-04LX laser range finder [3]. Then, the obtained map is used to plan the motion of the robot. The methods presented in this paper increase the efficiency of the motion planning method. With the recent improvements the robot is capable to climb higher obstacles by avoiding deadlocks. The deadlocks are caused by the lack of static stability or the lack of a kinematic margin. The results are obtained using various optimization methods to find a path of feet during swing phase, to optimize posture of the robot and to smooth the obtained path.

1.1. Related work

Autonomous walking on rough terrain is much more challenging than indoor locomotion [20]. Many impressive results were obtained using the LittleDog robot [6, 19, 32]. In this case the robot also uses various optimization methods during motion planning. The trajectory of the robot is optimized to maximize the stability and kinematic reachability. In the work of Ratliff *et al.* [19] the trajectory optimization is performed using the gradient-based Covariant Hamiltonian Optimization and Motion Planning (CHOMP) algorithm. Similarly, Kalakrishnan *et al.* proposed the use of optimization methods to improve the trajectory of the robot's body with respect to the initial Zero Moment Point (ZMP) trajectory [7]. To plan the motion of the robot the Anytime Repairing A* (ARA*) algorithm [18] is used. The algorithm maximizes footholds

rewards from the initial to the goal position.

Most of the walking robots as well as control methods are inspired by biological systems. Inspirations for six-legged robot are taken from behavior of stick insects [15]. The robot can use a dynamic gait which is based on results of Nature observations [30]. Periodic signals to control gait of the robot can be obtained using a Central Pattern Generator (CPG). The concept of CPGs is based on the real nervous structure which can be found in living creatures [31]. This concept is also successfully applied in multi-legged walking robots [9].

In our research we also take inspiration from biology. Our robot uses statically stable gaits and the leg coordination patterns, as stick insect do. It walks on rough terrain using secure and rather slow type of locomotion. The control system of the Messor robot strongly depends on the exteroceptive sensory system. Dynamic control techniques require high-torque drive units and precise proprioceptive sensing [14], which are not available to the Messor robot walking on rough terrain. Thus, the dynamic capabilities of our robot are limited.

Not only optimization and inspirations from the Nature can be used to obtain control system for walking robot. The robot can also learn how to walk. In [26] a modified Reinforcement Learning algorithm is presented. It was shown that the robot can utilize past experience and faster learn efficient locomotion on flat terrain. Similarly, a gait of humanoid robot can be optimized to obtain maximal linear velocity [27].

Some procedures for control of six-legged robots, which allow them to climb high, but structural obstacles like steps [5] and stairs [25] are known from the literature. However, in our work we don't assume particular shapes of the obstacles. The robot autonomously can deal with various obstacle. This behavior is obtained using RRT-based planner and optimization methods which support motion planning.

In our work we utilize hybrid deliberative/reactive paradigm. When the initial path of the robot is planned the robot can execute the planned motion using reactive and fast controller which guarantees more robust locomotion. Moreover, during execution of the planned motion the robot uses simple leg compliant strategy. The robot stops the leg's motion whenever contact force at leg's tip exceeds given thresholds. The compliant strategy is very important to robust locomotion [10, 31] or motion of a robot arm [13, 28].

2. Motion planning algorithm

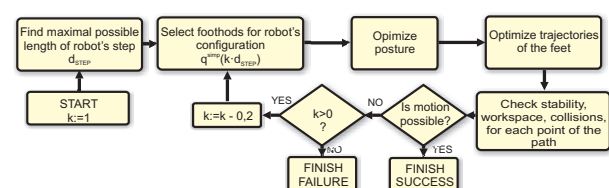


Fig. 2. EXTEND procedure used for RRT-based motion planning

The RRT-Connect-based integrated motion planner [2] creates two trees, which are extended alternately. The root of the first tree is located in the initial position of the robot. The root of the second tree is located in the goal position of the robot. At the beginning the algorithm tries to add new node to the first tree (EXTEND procedure). The position of the new node is determined randomly. If the robot can reach the next goal position $q(k \cdot d_{\text{STEP}})$ the node is added to the tree. Then, the second tree is extended in the direction to the new node. In the following iteration of the algorithm the sequence is swapped and the second tree is extended in the random direction at the beginning.

The core of the RRT algorithm is the EXTEND procedure (Fig. 2). The procedure checks if the transition in the desired direction is possible. If the motion is possible the procedure plans the precise path for the body and each foot. The previous version of the procedure has been modified [2]. First, the new procedure determines the maximal step length in the desired direction (d_{STEP}). The maximal step length is found using kinematic and collision constraints. To determine the maximal length of the step we create the initial posture of the robot q_{simp} . The procedure assumes initial horizontal orientation of the robot's body. The distance to the ground is set to such a value that guarantees secure clearance between robot's body and the ground (in experiments the clearance is set to 0.1 m). The sub-procedure, which plans the path for the next robot step, is executed five times for various step lengths $k \cdot d_{\text{STEP}}$ ($k \in \{0.2; 0.4; \dots; 1.0\}$). For the next potential node of the tree the robot finds footholds and optimizes the posture [3, 4]. Next, the procedure plans path of feet during swing phase. The path of the body is straight between two neighboring nodes. Then, the procedure checks if the execution of the planned path is possible. Workspaces of the robot's feet, collisions and static stability are checked. The procedure returns the planned path for the maximal possible step length.

2.1. Posture optimization strategy

The RRT algorithm determines a new (x_R, y_R) position of the robot in the global coordinate system. For this position the planning procedure determines full state (posture) of the robot (vertical position of the robot's platform z_R , inclination and leg's configuration). To this end, the elevation map of the environment is used. The direction of motion is represented by the vector $\overrightarrow{\mathbf{p}_R^i \mathbf{p}_R^{i+1}}$ identifying the previous position of the robot $\mathbf{p}_R^i = [x_R^i, y_R^i]$ and the next position of the robot $\mathbf{p}_R^{i+1} = [x_R^{i+1}, y_R^{i+1}]$. The orientation of the robot on the surface γ_R is equal to the angle between the vector $\overrightarrow{\mathbf{p}_R^i \mathbf{p}_R^{i+1}}$ and x axis of the global coordinate system. If the desired rotation around z axis can't be executed with the given kinematics of the robot (the reference angle γ_R is not reachable) the reference rotation angle γ_R is limited to maximal rotation which can be executed by the robot in a single step. In all experiments presented in the paper the maximal rotation angle is set to 0.2 rad.

For the new position $\mathbf{p}_R^{i+1} = [x_R^{i+1}, y_R^{i+1}]$ the robot

selects footholds [3]. Inclination of the robot's platform (θ_R, φ_R) and distance to the ground z_R are determined by the posture optimization procedure.

The optimization procedure searches for the vector $\mathbf{p}_R = [\theta_R, \varphi_R, z_R]^T$ which determines the posture of the robot with maximal kinematic margin of the robot d_{KM} :

$$\arg \max_{\mathbf{p}_R} \{d_{\text{KM}}(\mathbf{q}(\mathbf{p}_R))\}, \mathbf{q}(\mathbf{p}_R) \in C_{\text{free}}, \mathbf{q}(\mathbf{p}_R) \in C_{\text{stab}}, \quad (1)$$

where:

$\mathbf{q}(\mathbf{p}_R)$ – posture of the robot for the given configuration \mathbf{p}_R and position of feet determined using foothold selection algorithm,

$d_{\text{KM}}(\mathbf{q}(\mathbf{p}_R))$ – kinematic margin of the robot,

C_{free} – configurations of the robot which are collision free (lack of collisions with the ground and between parts of the robot) and inside of the robot's workspace,

C_{stab} – statically stable configurations of the robot.

To find the optimal posture of the robot the Particle Swarm Optimization (PSO) algorithm is used [4, 12]. The kinematic margin d_{KM}^i of the i -th leg is defined as the distance from the current position of the feet to the boundary of the reachable area of the leg [17]. To compute kinematic margin d_{KM} of the robot the distance from current positions of the feet to the boundaries of the legs workspace are computed. The smallest distance ($\min(d_{\text{KM}}^{i=1}, \dots, d_{\text{KM}}^{i=6})$) determines the d_{KM} value [4]. The computation of the kinematic margin should be fast. It is important because the kinematic margin is computed hundred of times during single optimization run. The analytical relation between configuration of the leg and the value of the kinematic margin is found to speed up the computations [4]. To obtain this relation, the approximation with mixture of Gaussian functions is used.

2.2. Leg-end optimization during swing phase

When the posture of the robot is determined the robot plans the path of each foot from the current to the next foothold. The initial paths are located on the plane which is perpendicular to the gravity vector. The initial and the goal footholds are located on this plane. The shape of the initial path is created according to the vertical cross-section of the obstacles between footholds [2]. The safety margin is added to decrease the risk of collision. However, the found path does not guarantee proper execution of the planned motion. Some points of the planned path might be unreachable for the robot (the positions of feet are outside of the robot workspace). Moreover the legs of the robot might collide with other parts of the robot. Thus, it is necessary to modify the position of feet in the direction perpendicular to the direction of the foot motion. We call it leg-end optimization during swing phase. During the optimization the inverse and forward kinematic of the leg are known, thus we can switch easily between the leg configuration and position of the foot.

We use both representations in the optimization procedure. For example we use leg configuration to compute kinematic margin and we use position of the foot to check collisions with the ground.

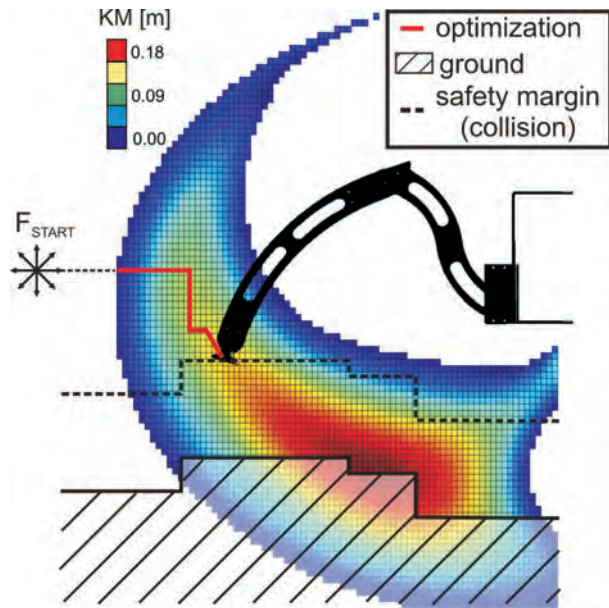


Fig. 3. Cross-section of the leg workspace and progress of the gradient-based optimization method during swing phase

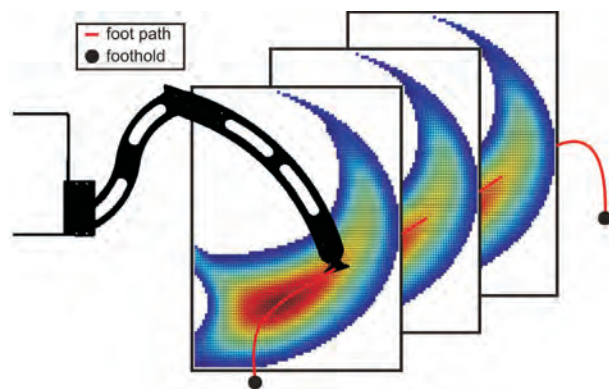


Fig. 4. Optimization of the leg-end position during swing phase for the following points of the initial path

The goal of the optimization (in our case sub-optimization) is to find the position of the i -th foot during swing phase $\mathbf{p}_f^i = [x_f^i, y_f^i, z_f^i]^T$ (expressed in the global coordinate system O) which satisfies the requirements for the value of the i -th leg kinematic margin d_{KM}^i :

$$d_{KM}^i(\mathbf{p}_f^i) > d_{KM}^{safe}, \mathbf{p}_f^i \in C_{free}, \quad (2)$$

where C_{free} are leg's configurations which are collision free and inside of the workspace of the robot and $d_{KM}^{safe} = 8 \text{ cm}$.

We decided to use sub-optimization instead of full optimization because looking for leg-end position which satisfies safety requirements is sufficient. The sub-optimization stops searching when the kinematic

margin is bigger than $d_{KM}^{safe} = 8 \text{ cm}$. This approach speeds up the optimization procedure. The procedure is run for each leg-end position and all points of the initial path during swing phase (Fig. 4).

To find the optimal foot position the method of steepest descent is used [11]. The cross-section of the leg workspace and progress of the gradient-based optimization method during swing phase are shown in Fig. 3. If the initial position of the foot F_{START} is outside of the workspace of the leg the robot searches for the acceptable leg's configuration (reference values for servomotors) which has positive value of the kinematic margin. To this end, the current position of the foot is moved iteratively in eight main directions (Fig. 3). When the foot is inside of the workspace of the robot, the procedure continues with gradient-based optimization. The position of the foot is modified in the direction perpendicular to the direction of the foot motion (vertical cross-section through the leg's workspace). The search space is limited by the shape of the obstacles increased by the safety margin (in the simulations presented in this paper the safety margin is 3 cm).

2.3. Path smoothing

The final path found by the RRT-based algorithm is made from straight lines which connect optimal postures of the robot. Thus, the motion of the robot is not smooth. In the nodes of the tree (positions of the robot where all feet are located on the ground and optimal position is determined) the robot rapidly changes the direction of the motion. To avoid rapid movements we use path smoothing on the results obtained from the RRT-based planner [8].

The goal of the path smoothing is the minimization of fitness function F_{sm} . The arguments of F_{sm} are the positions of points in the initial path \mathbf{p} found by the RRT-based algorithm, and positions of the points in the smooth path \mathbf{p}' :

$$\arg \min_{\mathbf{p}'} F_{sm}(\mathbf{p}') = \sum_{i=0}^n \left\{ \frac{\alpha}{2} (p_i - p'_i)^2 + \frac{\beta}{2} (p'_i - p'_{i-1})^2 + \frac{\beta}{2} (p'_i - p'_{i+1})^2 \right\}. \quad (3)$$

The element of the fitness function F_{sm} which depends on the β coefficient is responsible for path smoothing. Additionally, the element which depends on the α coefficient is introduced. Without this element the optimal solution is a straight line (the positions of the initial and final points are not optimized). To find the optimal value of the function (3) the gradient-based optimization is used [11]. At the beginning of the optimization the smoothed path \mathbf{p}' is the same as the initial path \mathbf{p} . In the following step of the optimization the algorithm modifies the position of the new path according to the equation:

$$p_i^{new} = p'_i + \alpha(p_i - p'_i) + \beta(p'_{i-1} + p'_{i+1} - 2p'_i). \quad (4)$$

The positions of footholds are not modified by the optimization. Similarly, the points of the path are not modified if the new position is outside of the robot workspace, if it causes collisions, or the new position of the robot is not statically stable. The procedure ends when the new modification of the path Δ_p is below the pre-determined value Δ_{max} :

$$\Delta_p = \sum_{i=0}^n |p_i^{new} - p'_i| < \Delta_{max}. \quad (5)$$

3. Results

The results of the procedure which optimizes the position of feet during swing phase is presented in Fig. 5. During the experiment the robot was climbing a step. When the optimization is used the robot modifies the initial path of feet. Without using the proposed method the desired position of foot is outside of the robot's workspace. The execution of the planned motion is not possible. When the proposed optimization procedure is used the robot moves its leg sideway to increase kinematic margin to avoid collisions with the obstacle. This optimization procedure allows to climb higher obstacles and avoid deadlocks.

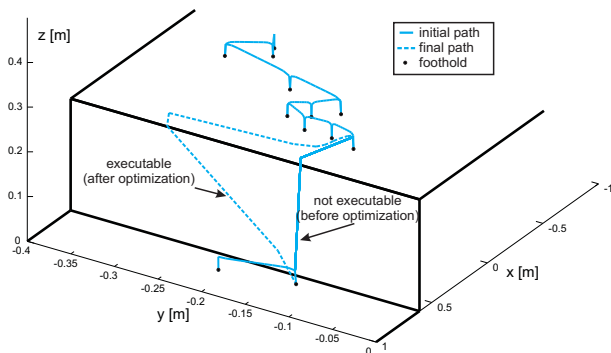


Fig. 5. Results of the foot path optimization procedure during swing phase on the step obstacle

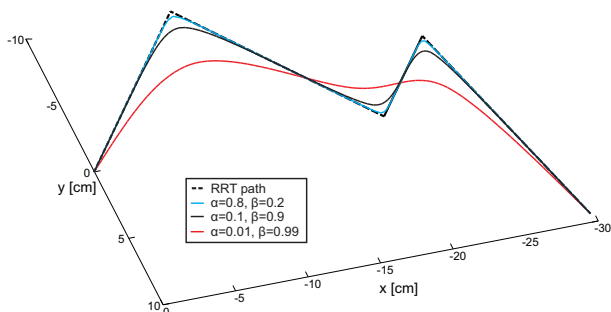


Fig. 6. Results of the path smoothing procedure for various α and β parameters

Before execution of the planned motion the path is smoothed. The sequence of the body positions $[x_R, y_R, z_R, \theta_R, \varphi_R, \gamma_R]$ as well as path of each foot is modified. Example results of path smoothing for various values of parameters are shown in Fig. 6. When the

parameter β is increased and the parameter α is decreased the obtained path is smoother. When parameter α is increased and parameter β is decreased the obtained path is similar to the initial path (Fig. 6).

The parameters are set to $\alpha = 0.01$ and $\beta = 0.99$ when the path for the body is smoothed. The parameters are set to $\alpha = 0.8$ and $\beta = 0.2$ when the paths of feet are smoothed. The goal is to obtain smooth path for the robot's body to limit unwanted sudden motions. The paths of feet are more constraint (workspace of the robot and collisions with obstacles). The footholds can't be modified in this stage thus the path is only slightly smoothed.

The proposed methods were verified in the realistic simulator of the Messor walking robot [24]. Dynamic properties of the robot are simulated using the Open Dynamics Engine [22]. The software library detects collisions between parts of the robot's body and simulates behavior of the colliding objects (static and dynamic friction). The simulator also contains the software controller of the real robot. The environment, the state of the robot, and paths obtained using the motion planning algorithm are presented using OpenGL procedures.

The properties of the proposed methods were presented in simulations using three various obstacles¹: a bump (Fig. 7a), concavity (Fig. 7b) and step (Fig. 7c). Using standard shape of the obstacles we can present efficiency of the proposed methods. We found the maximal size of the obstacles which can be covered by the robot using various variants of the motion planning approach (Table 1). It should be also mentioned that the proposed method is universal and does not assume any particular model of the obstacle.

Tab. 1. Maximal size of the obstacles which can be covered by the robot using various variants of the motion planning approach. Table presents three variants: motion planning without optimization (1), with posture optimization (2) and with all optimization procedures (3).

variant	bump [m] Fig. 7a	concavity [m] Fig. 7b	step[m] Fig. 7c
1	0.13	-0.17	0.15
2	0.16	-0.185	0.18
3	0.16	$-\infty$	0.25

To show the efficiency of using presented optimization procedures we performed simulations with three different obstacles: a bump, a concavity and a step. In the first simulation the robot plans its motion using only the RRT algorithm. The position of the robot's platform is determined using a simple strategy – the orientation of the robot's body is the same as the average slope of the terrain. The distance to the ground is set to a value which is minimal but safe and guarantees that the motion is collision free [2]. In the second simulation the robot uses posture optimization algorithm to determine its posture in the nodes of the RRT trees. In the third simulation the robot uses all methods presented in this paper (posture optimization, feet path planning during swing phase and path

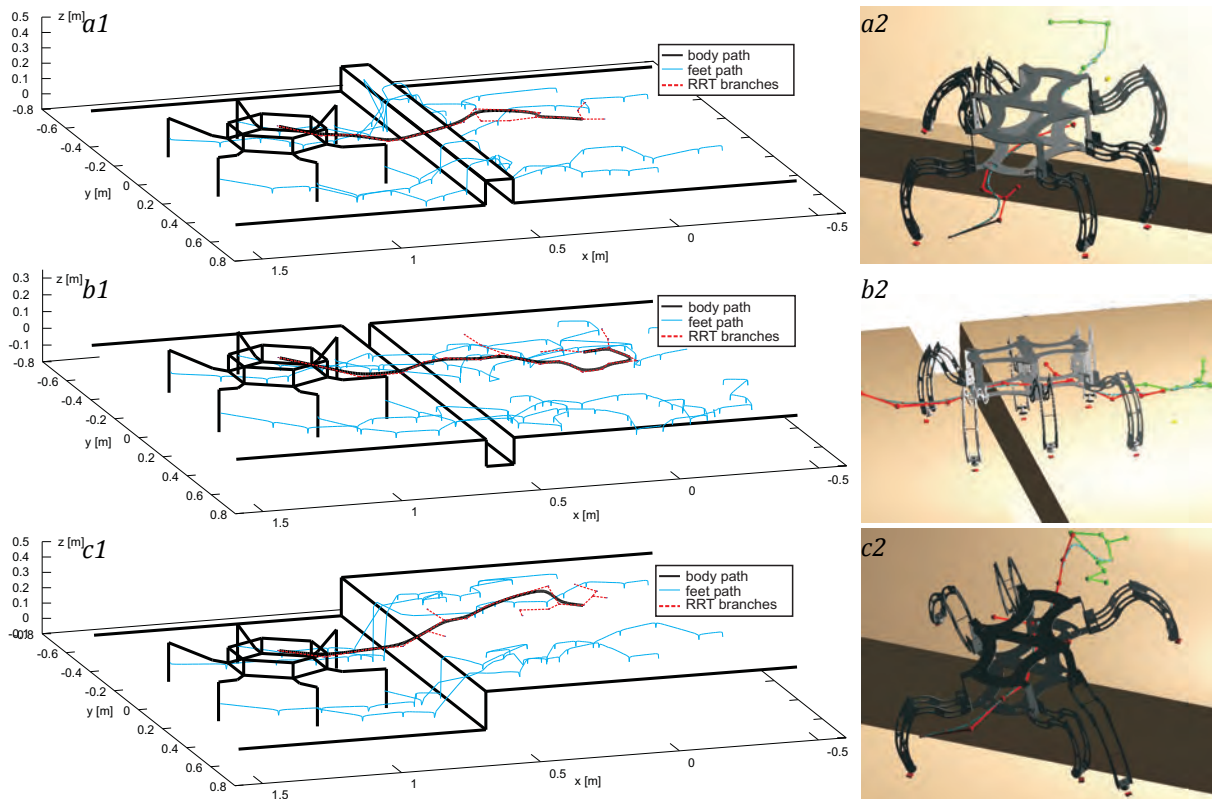


Fig. 7. The planned path of the robot (1) and execution of the planned motion in simulator (2) while climbing: a – bump, b – concavity, c – step

smoothing).

When the robot does not use optimization methods the maximal height of the bump which can be covered by the robot is 13 cm (Tab. 1). Using posture optimization procedure the robot is capable to climb a bump which is 3 cm higher. The maximal height of the obstacle is the same when the optimization during swing phase is used. To deal with the obstacle the robot has to put its legs on the ground level and on the obstacle at the same time. If the obstacle is too high the reference position of the feet are outside of the workspace of the robot. The optimization during swing phase does not bring additional advantages. The path obtained using all optimization methods is shown in Fig. 7a.

When the posture optimization is used during motion planning the maximal depth of the concavity is 18.5 cm. If necessary the robot decreases the distance to the ground to reach the bottom of the concavity and safely moves to the other side of the ditch. When all optimization methods are used the depth of the concavity does not play any role. The EXTEND procedure (Fig. 2) checks various step lengths. For the given width of the ditch the robot does not have to place its feet on the bottom of the concavity. The robot can make longer steps and place feet on the other side of the concavity. The EXTEND procedure proposed in this paper allows to autonomously modify length of steps to deal with various obstacles. In this case path optimization during swing phase does not bring additional advantages. The initial path is secure and guarantees a proper kinematic margin.

The advantages of using optimization procedure during the swing phase can be observed during step climbing. The obtained path is shown in Fig. 7c. The motion of feet is modified to increase kinematic margin during swing phase. As a result the robot moves feet sideways when they are above the edge of the step. The proposed strategy allows to deal with steps 25 cm high. If only the posture optimization procedure is used the maximal height of the step is 18 cm (Tab. 1).

To check the efficiency of the algorithm on terrain with irregular obstacles a simulation on rough terrain mockup was performed. The results are presented in Fig. 8. The distance between initial and final position of the robot is 3 m. Because the algorithm is random-based, the series of 10 simulations was performed. In each trial the robot was able to find a secure path to the goal position. The average time of the path planning is 389 s and standard deviation is 44 s (simulations were performed on a computer with Intel i7 2.7GHz processor).

4. Conclusions

The article presents various optimizations strategies to plan motion of the six-legged walking robot on rough terrain. The PSO optimization algorithm is used to find posture of the robot during phase when all feet of the robot are placed on the ground. The gradient-based optimization method is used to determine feet position during swing phase. A similar method is used for smoothing the path returned by RRT-based motion planner. The presented methods are used as core



Fig. 8. Planned path during simulation on rough terrain mockup

elements of the new EXTEND procedure which determines the following step of the robot.

The goal of using the optimization methods is to create algorithm which is capable to plan motion of the robot on rough terrain with high obstacles. The capabilities of the algorithm were presented in the simulator. We determined the maximal size of the obstacles which can be covered by the robot using the presented methods. The goal of the optimization can be easily modified. It could be the maximal security margin (a sum of three weighted coefficients: value of distance between feet and obstacles, kinematic margin and stability margin) or combination of few coefficients. In the future we are going to implement the proposed optimization methods on the real Messor robot. To this end, we are going to integrate the mapping algorithm which uses Hokuyo laser range finder and a self-localization algorithms.

5. Acknowledgement

This work was supported by NCN grant no. 2011/01/N/ST7/02080.

The author would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Notes

¹A video with simulation experiments is available on <http://irm.cie.put.poznan.pl/jamris2013.wmv>

AUTHOR

Dominik Belter* – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Dominik.Belter@put.poznan.pl.

*Corresponding author

REFERENCES

- [1] G. B. Bell, M. Livesey, "The Existence of Local Minima in Local-Minimum-Free Potential Surfaces", *Towards Autonomous Robotic Systems* (TAROS 2005), London, UK, 2005 pp. 9–14.
- [2] D. Belter, P. Skrzypczyński, *Integrated motion planning for a hexapod robot walking on rough terrain*, 18th IFAC World Congress, Milan, Italy, 2011, pp. 6918–6923
- [3] D. Belter, P. Skrzypczyński, "Rough terrain mapping and classification for foothold selection in a walking robot", *Journal of Field Robotics*, vol. 28(4), 2011, pp. 497–528
- [4] D. Belter, P. Skrzypczyński, "Posture optimization strategy for a statically stable robot traversing rough terrain". In: *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Villamoura, Portugal, 2012, pp. 2204–2209,
- [5] S. Fujii, K. Inoue, T. Takubo, T. Arai, "Climbing up onto Steps for Limb Mechanism Robot "ASTER-ISK"", In: *Proc of the 23rd International Symposium on Automation and Robotics in Construction*, 2006, pp. 225–230
- [6] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, "Fast, robust quadruped locomotion over challenging terrain". In: *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010, pp. 2665–2670
- [7] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain", *Int. Journal of Robotics Research*, vol. 30(2), 2010, pp. 236–258.
- [8] D. Dolgov, S. Thrun, "Detection of principal directions in unknown environments for autonomous navigation", In: *Proc. of the Robotics: Science and Systems*, 2008
- [9] A. Gmerek, "Wykorzystanie quasi-chaotycznych oscylatorów do generowania rytmu chodu robotów kroczących", *Postępy robotyki* 2012, K. Tchoń, C. Zieliński (Eds.).
- [10] A. Gmerek, E. Jezierski, "Admittance control of a 1-DoF robotic arm actuated by BLDC motor". In: *17th International Conference on Methods and Models in Automation and Robotics* (MMAR), 2012, pp. 633–638.
- [11] C.T. Kelley, *Iterative methods for optimization*, SIAM, Philadelphia, 1999.
- [12] J. Kennedy, R.C. Eberhart, "Particle swarm optimization". In: *Proc. IEEE Int. Conf. on*

- Neural Networks*, Piscataway, Australia, 1995, pp. 1942–1948.
- [13] M. Kordasz, R. Madoński, M. Przybyła, P. Sauer, *Active Disturbance Rejection Control for a Flexible-Joint Manipulator*, Lecture Notes in Control and Information Sciences, Robot Motion and Control 2011, K. Kozłowski (Ed.), Springer-Verlag, 2011, pp. 247–256.
- [14] M. Kowalski, M. Michalski, D. Pazderski, *Quadruped walking robot WR-06 - design, control and sensor subsystems*, Lecture Notes in Control and Information Sciences, Robot Motion and Control 2009, Springer-Verlag, 2009, pp. 175–184.
- [15] S. Krenich, M. Urbanczyk, "Six-Legged Walking Robot for Inspection Tasks", *Solid State Phenomena*, vol. 180, pp. 137–144, 2011
- [16] J.J. Kuffner, S.M. LaValle, "RRT-Connect: An efficient approach to single-query path planning". In: *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, USA, 2000, pp. 995–1001.
- [17] P. D. Kumar, D. Kalyanmoy, and G. Amitabha, "Optimal path and gait generations simultaneously of a six-legged robot using a GA-Fuzzy approach", *Robotics and Autonomous Systems*, vol. 41, no. 1, 2002, pp. 1-20.
- [18] M. Likhachev, G. Gordon, S. Thrun, "ARA*: anytime A* with provable bounds on suboptimality". In: *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 conference*, 2003.
- [19] N. Ratliff, M. Zucker, J.B. Andrew, S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning". In: *Proc. IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, 2009, pp. 489– 494.
- [20] R.B. Rusu, A. Sundaresan, B. Morisset, K. Hauser, M. Agrawal, J.C. Latombe, M. Beetz, "Leaving flatland: efficient real-time 3D perception and motion planning", *Journal of Field Robotics*, vol. 26(10), 2009, pp. 841–862.
- [21] U. Saranli, M. Buehler, D.E. Koditschek, "RHex: a simple and highly mobile hexapod robot", *International Journal of Robotics Research*, vol. 20, 2001, pp. 616–631.
- [22] R. Smith, *Open Dynamics Engine*, <http://www.ode.org>, 2012.
- [23] K. Walas, D. Belter, "Supporting locomotive functions of a six-legged walking robot", *International Journal of Applied Mathematics and Computer Science*, vol. 21, no.2, 2011, pp. 363–377.
- [24] K. Walas, D. Belter, "Messor – Versatile walking robot for search and rescue missions", *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 5, no. 2, 2011, pp. 28–34.
- [25] K. Walas, A. Kasiński, "Controller for Urban Obstacles Negotiation with Walking Robot". In: *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Villamoura, Portugal, 2012, pp. 181–186.
- [26] P. Wawrzyński, "Real-Time Reinforcement Learning by Sequential Actor-Critics and Experience Replay", *Neural Networks*, vol. 22, no. 10, Elsevier, 2009, pp. 1484-1497.
- [27] P. Wawrzyński, "Autonomous Reinforcement Learning with Experience Replay for Humanoid Gait Optimization", *Proceedings of the International Neural Network Society Winter Conference*, Procedia, 2012, pp. 205–211.
- [28] T. Winiarski, A. Woźniak, "Indirect force control development procedure", *Robotica*, vol. 31, 4, pp. 465–478, 2013.
- [29] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A.A. Rizzi, "Raibert, Autonomous navigation for BigDog". In: *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010, pp. 4736–4741
- [30] T. Zielińska, "Biological inspiration used for robots motion synthesis", *Journal of Physiology – Paris*, vol. 103(3-5), September 2009, pp. 133–140, 2009.
- [31] T. Zielińska, A. Chmielniak, "Biologically inspired motion synthesis method of two-legged robot with compliant feet", *Robotica*, vol. 29, no. 7, 2011, pp. 1049–1057.
- [32] M. Zucker, J.A. Bagnell, C.G. Atkeson, J. Kuffner, "An optimization approach to rough terrain locomotion". In: *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010, pp. 3589–3595