Footstep Planner Algorithm for a Lower Limb Exoskeleton Climbing Stairs

Sergey Jatsun, Sergei Savin(✉), and Andrey Yatsun

Southwest State University, Kursk, Russia
savinswsu@mail.ru

Abstract. In this paper, a footstep planning algorithm for a lower limb exoskeleton climbing stairs is presented. The algorithm relies on having a height map of the environment, and uses two procedures: partial decomposition of the supporting surface into convex obstacle-free regions, and optimization of the foot step position implemented as a quadratic program. These two methods are discussed in detail in the paper, and the simulation results are shown. It is demonstrated that the algorithm works for different staircases, and even for the staircases with obstacles on them.

Keywords: Footstep planner algorithm · Lower limb exoskeletons · Quadratic programming · Climbing stairs

1 Introduction

Exoskeletons have been a focus of many studies in robotics for the past few decades because of the significant range of possible practical applications that this technology has. This variety of applications has already lead to a significant interest in the industry, resulting in a number of successfully working exoskeleton designs [1–3]. The lower limb exoskeletons are an especially important example of this development. They can serve to restore mobility to patients, and enable workers and soldiers to perform more physically demanding tasks [3].

At the same time, there is still a number of challenging problems for exoskeletons. One of the most important examples of such a problem is climbing stairs [4]. The challenges associated with climbing stairs include the problem of maintaining vertical balance of the mechanism, higher demands on the mechanical design and motors (as compared to walking on a horizontal plane) and the footstep planning problem [5, 6]. Here we consider the latter. A footstep planning algorithm should be able to produce a feasible sequence of steps using the information about the exoskeleton's position and the environment. The problem of processing the sensory data in order to construct a map of the environment, and find robot's location on it, is closely related to footstep planning [7]. This problem is usually being solved before the footstep planning algorithm can start working and is often associated with simultaneous localization and mapping methods or constructing the so-called height map [7, 8]. In this paper, we assume that the robot has access to the height map of the environment, and we focus on using this information in order to plan a sequence of steps.

A. Ronzhin et al. (Eds.): ICR 2017, LNAI 10459, pp. 75–82, 2017.
DOI: 10.1007/978-3-319-66471-2_9

There are a few different approaches to footstep planning. In the paper [9] an optimization-based footstep planner is discussed. One of its features is the use of mixed-integer programming for planning and IRIS algorithm for the decomposing the supporting surface into convex regions (see [10] for details). Alternative approach based on stereographic projection is shown in [11]. In [12, 13] a footstep planner was realized using quadratic programming (QP) for different types of walking robots. In [14] a footstep planner for Honda's biped robot was designed based on the A* search algorithm. This planner does not guarantee optimality of the chosen path, but works at a satisfactory speed. We should note that significant progress in numerical techniques for solving quadratic programs has been made over the last decades, making the use of QP in real time control loops a viable solution. Examples of this can be found in works [15, 16]. In this paper, we use also use quadratic programming as a part of the footstep planning algorithm. Our algorithm also requires partial decomposition of the supporting surface (or its height map) into convex regions. The algorithm used for this decomposition is described in the following chapters.

2 Exoskeleton Description

In this paper, we focus on a footstep planner algorithm for a lower limb exoskeleton. We consider the exoskeleton ExoLite, which has two legs divided into four segments (hip, thigh, shin and foot). This exoskeleton has 10 actuated joints, one in each hip, thigh, and knee, and two in each foot [17–23]. Each joint is equipped with sensors for measuring joint angles and the exoskeleton's feet are additionally equipped with pressure sensors and inertial measurement units. A general view of the exoskeleton is shown in Fig. 1.

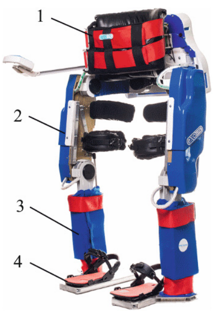

Fig. 1. General view of the ExoLite exoskeleton; 1 – torso link, 2 – thigh link, 3 – shin link, 4 – foot link

In papers [17–19] the task of controlling the robot while maintaining its vertical balance is discussed. Papers [20–23] are studying the problem of controlling the exoskeleton during the execution of prescribed trajectories, as well as the problem of tuning the controller.

The contact surface of the exoskeleton's feet has the form of a non-convex polygon. In this study, we consider an algorithm that finds a new placement position for the exoskeleton's feet. The new position should guarantee that the foot lies inside an obstacle-free region on the supporting surface. If we only consider obstacle-free regions that are convex, then we can modify this condition in the following way: the convex hull F_c of the points on the exoskeleton's foot should lie inside an obstacle-free region on the supporting surface. To show the equivalence of the two conditions we can observe that if all points on the exoskeleton's foot lie inside a convex region, then any convex combination of these points also lies inside that region, which in turn means that their convex hull lies inside the mentioned region.

In the following chapters, we assume that the robot has access to the height map of the supporting surface. The height map is a function $z_{map} = z_{map}(x, y)$ that returns a height z for a given point on the supporting surface with coordinates x and y.

3 Supporting Surface Partial Decomposition Algorithm

In this chapter, we consider an algorithm that we use to partially decompose the surface into convex obstacle-free regions. By obstacle-free region we mean a region where the heights of any two points are different by no more than ε, the height variation threshold. We say that the surface is partially decomposed to indicate that the procedure is meant to provide us with a few convex obstacle-free regions, but these regions do not need to tile the whole surface.

The first step of the algorithm is to generate a set of seed points $\Sigma = \{\sigma_i\}$, where $\sigma_i = [\, x_i \quad y_i \,]$ is a seed point with Cartesian coordinates x_i and y_i. To generate the set Σ we can use a grid, a set of random numbers or a low-discrepancy sequence of points, such as the Sobol sequence [24].

The second step is to take one seed point σ_i and construct an approximation of the convex obstacle-free region it belongs to. This is done in the following way. First we construct a sequence of n rays that originate from the point σ_i. The first ray can be chosen randomly, and every other ray in the sequence is constructed by rotating the previous one by $2\pi/n$ radians. We can parametrize these rays by a parameter ξ that represents the distance from the seed point σ_i. For the case when the first ray is chosen to lie along the positive x axis on the xy plane every point on each ray can be described by the next formula:

$$\chi_{i,j}(\xi) = \sigma_i + [\, \xi \cos(2j\pi/n) \quad \xi \sin(2j\pi/n) \,], \tag{1}$$

where $\chi_{i,j}$ is a point on the j-th ray and ξ is a variable that determines the distance from $\chi_{i,j}$ to the seed point σ_i. Then for each ray we find a point $v_{i,j}$ (where i is the index number of the seed point, and j is the index number of the ray, $j = \overline{1, n}$) closest to point σ_i out of all points on the ray which lie outside the obstacle-free region. His can be formulated as follows:

$$\begin{cases} v_{i,j} = \chi_{i,j}(\xi_{\min}) \\ \xi_{\min} = \min\{\xi : \left| z_{map}(\chi_{i,j}(\xi)) - z_{map}(\sigma_i) \right| > \varepsilon/2\} \end{cases}, \tag{2}$$

where $\xi_{\min}$ is the minimum value of ξ, such that the point $\chi_{i,j}$ lies on an obstacle. To determine whether or not $\chi_{i,j}(\xi)$ lies on an obstacle we use the height map z_{map} to check if the height difference between point $\chi_{i,j}$ and the seed point σ_i is greater than $\varepsilon/2$, which gives us a conservative estimation.

Thus we can construct an approximation of the convex obstacle-free region as a convex hull of the points $v_{i,j}$ (for a given i and all j). We denote this region as $\boldsymbol{\Omega}_i$:

$$\boldsymbol{\Omega}_i = \mathrm{Conv}\{v_{i,j}\} \quad j = \overline{1, n}. \tag{3}$$

We should note that the quality of this approximation depends the shape of the actual obstacle free region and on the number of rays n. If the region is convex, then as the number of rays grows the approximation will approach the exact shape of the region.

Because $\boldsymbol{\Omega}_i$ is a convex hull of a finite number of points it is a polygon with vertices $v_{i,j}$, and as any convex polygon it can be represented as a set of linear inequalities:

$$\boldsymbol{\Omega}_i = \{\mathbf{r} : \mathbf{A}_i \mathbf{r} \le \mathbf{b}_i\}, \tag{4}$$

where $\mathbf{A}_i$ and $\mathbf{b}_i$ are a matrix and a vector that correspond to a linear inequality representation of $\boldsymbol{\Omega}_i$, and $\mathbf{r}$ is a radius vector for a point in the polygon $\boldsymbol{\Omega}_i$. For the discussion of algorithms for computing $\mathbf{A}_i$ and $\mathbf{b}_i$ from the given set of vertices $v_{i,j}$, see [25].

The third step of the algorithm is the improvement of the obstacle-free region approximation. Since $\boldsymbol{\Omega}_i$ was chosen as a convex hull of the points $v_{i,j}$, it can contain some of these points in its interior. During this step we add new inequalities to the system $\mathbf{A}_i \mathbf{r} \le \mathbf{b}_i$ and construct a new convex polygon $\boldsymbol{\Omega}_i^*$, such that every point $v_{i,j}$ is either outside $\boldsymbol{\Omega}_i^*$ or lie on its boundary.

Let us assume that a vertex defined by its radius vector $\mathbf{v}_{in}$ lies in the interior of $\boldsymbol{\Omega}_i$. Then, we can add the following linear inequality to the system $\mathbf{A}_i \mathbf{r} \le \mathbf{b}_i$ to make it lie on the boundary of a new polygon $\boldsymbol{\Omega}_i^*$:

$$\begin{cases} \mathbf{a}_{new}^{\mathrm{T}} \mathbf{r} \le b_{new} \\ \mathbf{a}_{new} = (\mathbf{v}_{in} - \boldsymbol{\sigma}_i)/\|\mathbf{v}_{in} - \boldsymbol{\sigma}_i\| \\ b_{new} = \|\mathbf{v}_{in}\| + \mathbf{a}_{new}^{\mathrm{T}} \boldsymbol{\sigma}_i \end{cases}, \tag{5}$$

where $\boldsymbol{\sigma}_i$ is the radius vector defining the position of the point σ_i. Then the new polygon $\boldsymbol{\Omega}_i^*$ is defined as follows:

$$\boldsymbol{\Omega}_i^* = \left\{ \mathbf{r} : \begin{bmatrix} \mathbf{A}_i \\ \mathbf{a}_{new}^{\mathrm{T}} \end{bmatrix} \mathbf{r} \le \begin{bmatrix} \mathbf{b}_i \\ b_{new} \end{bmatrix} \right\}. \tag{6}$$

We iteratively add new constraints to the polygon $\boldsymbol{\Omega}_i^*$ until all vertices $v_{i,j}$ are outside $\boldsymbol{\Omega}_i^*$ or lie on its boundary.

4 Quadratic Programming-Based Footstep Planning Algorithm

In this chapter, we discuss a footstep planning algorithm for climbing stairs based on quadratic programming. We denote as $\mathbf{r}_d = [x_d \quad y_d]^{\mathrm{T}}$ a desired step. A desired step is determined by the user, and the planning algorithm tries to find a feasible step that would be as close as possible to $\mathbf{r}_d$. We formulate it as the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \mathbf{r}_s^{\mathrm{T}}\mathbf{r}_s \\ \text{subject to} & \mathbf{A}_i^*(\mathbf{r}_d + \mathbf{r}_s + \mathbf{r}_{e,j}) \leq \mathbf{b}_i^* \end{array}, \tag{7}$$

where $\mathbf{r}_s = [x_s \quad y_s]^{\mathrm{T}}$ is a shift from the desired foot position $\mathbf{r}_d$, $\mathbf{r}_{e,j}$ is the relative position of the j-th vertex of the polygon F_c, which corresponds to a vertex of the exoskeleton's foot, and $\mathbf{A}_i^*$ and $\mathbf{b}_i^*$ are a matrix and a vector in the linear inequality representation of the obstacle-free region Ω_i^*. This problem is solved for all obstacle-free regions Ω_i^* that were found using the previously discussed decomposition procedure, and then the solution with the smallest value of the cost function is chosen.

This algorithm can be modified by changing the cost function to be a quadratic form of $\mathbf{r}_s$. Then by using an appropriate quadratic form matrix we can chose the preferred direction of the shift $\mathbf{r}_s$.

5 Simulation Results

In this chapter, we look at simulation results obtained using the presented algorithm. First, we consider two different staircases with pitch angles π/6 and π/12. Both staircases have a rise height of 0.12 m. The desired step length was chosen to be 0.32 m. The obtained footstep plans are shown in Fig. 2.

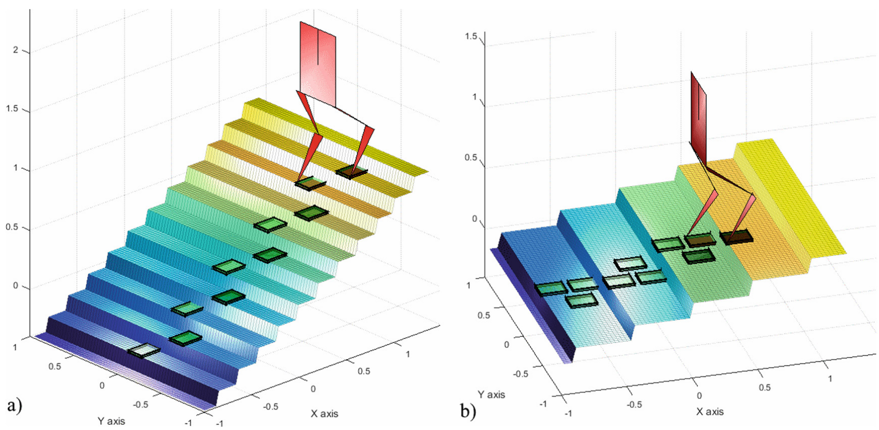

Fig. 2. The exoskeleton climbing staircases with pitch angles: (a) π/6, (b) x/12

We can note that for the steeper staircase (Fig. 2a) the footstep plan has hit every step of the staircase exactly once. For the other staircase, the footstep plan has hit every step three times. This happened because of the greater length of the steps. By doubling the desired step length, we can obtain a footstep plan that would skip every other step.

The proposed algorithm can also handle cases when there are additional obstacles on the staircase. Figure 3 shows the footstep plan generated for the case when the staircase has a pitch angle of $\pi/8$, and there are 3 obstacles lying on the stairs.

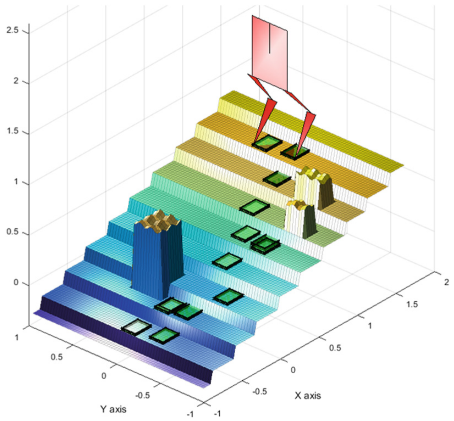

Fig. 3. The exoskeleton climbing a staircase with obstacles

We should note that the third step of the partial decomposition algorithm is mainly used to handle this type of problem. Stairs with no obstacles naturally provide convex regions, so only the first two steps of the decomposition algorithm are needed. On the other hand, addition of the third step to the decomposition algorithm allows to handle the problem of feet collision, by checking if the vertices of the stationary foot lie in the interior of the obstacle-free regions Ω_i^*, and augmenting Ω_i^* with new inequalities if they do.

6 Conclusions

In this paper, a footstep planning algorithm for climbing stairs was presented. The algorithm relies on having a height map of the supporting surface. It can be decomposed into two independent methods: first is the method for partial decomposition of the supporting surface into convex regions and the second is an algorithm for finding an optimal foot placement in these regions using quadratic programming.

It is shown that the footstep planning algorithm works for different staircases, and that it can handle the situation when there are obstacles lying on the stairs. It shows that the algorithm can be used as an alternative to the method presented in [26] for obstacle avoidance. The version of the algorithm presented in the paper plans for only one step ahead, but it is possible to extend it for planning multiple steps ahead.

Acknowledgements. Supported from grant of the President of the Russian Federation for young scientists MK-2701.2017.8.

References

1. Barbareschi, G., Richards, R., Thornton, M., Carlson, T. and Holloway, C.: Statically vs dynamically balanced gait: analysis of a robotic exoskeleton compared with a human. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6728–6731 (2015). doi:10.1109/EMBC.2015.73199372015
2. Kazerooni, H., Racine, J.L., Huang, L., Steger, R.: On the control of the berkeley lower extremity exoskeleton (BLEEX). In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 4353–4360 (2005). doi:10.1109/ROBOT.2005.1570790
3. Ferrati, F., Bortoletto, R., Menegatti, E., Pagello, E.: Socio-economic impact of medical lower-limb exoskeletons. In: 2013 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 19–26 (2013). doi:10.1109/ARSO.2013.6705500
4. Fu, C., Chen, K.: Gait synthesis and sensory control of stair climbing for a humanoid robot. IEEE Trans. Ind. Electron. **55**(5), 2111–2120 (2008). doi:10.1109/TIE.2008.9212052008
5. Lum, H.K., Zribi, M., Soh, Y.C.: Planning and control of a biped robot. Int. J. Eng. Sci. **37** (10), 1319–1349 (1999). doi:10.1016/S0020-7225(98)00118-9
6. Figliolini, G., Ceccarelli, M.: Climbing stairs with EP-WAR2 biped robot. In: IEEE International Conference on Robotics and Automation, Proceedings 2001 ICRA, vol. 4, pp. 4116–4121 (2001). doi:10.1109/ROBOT.2001.933261
7. Fallon, M., Kuindersma, S., Karumanchi, S., Antone, M., Schneider, T., Dai, H., D'Arpino, C.P., Deits, R., DiCicco, M., Fourie, D., Koolen, T.: An architecture for online affordance-based perception and whole-body planning. J. Field Robot. **32**(2), 229–254 (2015). doi:10.1002/rob.21546
8. Thrun, S., Leonard, J.J.: Simultaneous localization and mapping. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, pp. 871–889. Springer, Heidelberg (2008). doi:10.1007/978-3-540-75388-9_3
9. Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., Tedrake, R.: Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot. Auton. Robots **40**(3), 429–455 (2016). doi:10.1007/s10514-015-9479-3
10. Deits, R., Tedrake, R.: Computing large convex regions of obstacle-free space through semidefinite programming. In: Akin, H.L., Amato, N.M., Isler, V., Stappen, A.F. (eds.) Algorithmic Foundations of Robotics XI. STAR, vol. 107, pp. 109–124. Springer, Cham (2015). doi:10.1007/978-3-319-16595-0_7
11. Savin, S.: An algorithm for generating convex obstacle-free regions based on stereographic projection. In: Proceedings of the 13th Siberian Conference SIBCON-2017 (2017, in publishing)
12. Feng, S., Whitman, E., Xinjilefu, X., Atkeson, C.G.: Optimization-based full body control for the DARPA robotics challenge. J. Field Robot. **32**(2), 293–312 (2015). doi:10.1002/rob.21559
13. Savin, S., Vorochaeva, L.Y.: Footstep planning for a six-legged in-pipe robot moving in spatially curved pipes. In: proceedings of The 13th Siberian Conference SIBCON-2017 (2017, in publishing)

14. Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., Kanade, T.: Footstep planning for the Honda ASIMO humanoid. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 629–634 (2005). doi:10.1109/ROBOT.2005.1570188
15. Kuindersma, S., Permenter, F., Tedrake, R.: An efficiently solvable quadratic program for stabilizing dynamic locomotion. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2589–2594 (2014). doi:10.1109/ICRA.2014.6907230
16. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Trans. Control Syst. Technol. **18**(2), 267–278 (2014). doi:10.1109/TCST.2009.20179342010
17. Panovko, G.Y., Savin, S.I., Yatsun, S.F., Yatsun, A.S.: Simulation of exoskeleton sit-to-stand movement. J. Mach. Manuf. Reliab. **45**(3), 206–210 (2016). doi:10.3103/S1052618816030110
18. Jatsun, S., Savin, S., Yatsun, A., Gaponov, I.: Study on a two-staged control of a lower-limb exoskeleton performing standing-up motion from a chair. In: Kim, J.-H., Karray, F., Jo, J., Sincak, P., Myung, H. (eds.) Robot Intelligence Technology and Applications 4. AISC, vol. 447, pp. 113–122. Springer, Cham (2017). doi:10.1007/978-3-319-31293-4_10
19. Jatsun, S.F.: The modelling of the standing-up process of the anthropomorphic mechanism. In: Assistive Robotics: Proceedings of the 18th International Conference on CLAWAR 2015, pp. 175–182 (2015). doi:10.1142/9789814725248_0024
20. Jatsun, S., Savin, S., Lushnikov, B., Yatsun, A.: Algorithm for motion control of an exoskeleton during verticalization. In: ITM Web of Conferences, vol. 6, pp. 1–6 (2016). doi:10.1051/itmconf/20160601001
21. Jatsun, S., Savin, S., Yatsun, A.: Parameter optimization for exoskeleton control system using Sobol sequences. In: Parenti-Castelli, V., Schiehlen, W. (eds.) ROMANSY 21 - Robot Design, Dynamics and Control. CICMS, vol. 569, pp. 361–368. Springer, Cham (2016). doi:10.1007/978-3-319-33714-2_40
22. Jatsun, S., Savin, S., Yatsun, A.: Comparative analysis of iterative LQR and adaptive PD controllers for a lower limb exoskeleton. In: 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 239–244 (2016). doi:10.1109/CYBER.2016.7574829
23. Jatsun, S., Savin, S., Yatsun, A., Postolnyi, A.: Control system parameter optimization for lower limb exoskeleton with integrated elastic elements. In: Advances in Cooperative Robotics, pp. 797–805 (2017). doi:10.1142/9789813149137_0093
24. Sobol, I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki **7**(4), 784–802 (1967)
25. Bremner, D., Fukuda, K., Marzetta, A.: Primal-dual methods for vertex and facet enumeration. Discret. Comput. Geom. **20**(3), 333–357 (1998). doi:10.1007/PL000093891998
26. Jatsun, S., Savin, S., Yatsun, A.: Study of controlled motion of an exoskeleton performing obstacle avoidance during a single support walking phase. In: 20th International Conference on System Theory, Control and Computing (ICSTCC), pp. 113–118 (2016). doi:10.1109/ICSTCC.2016.7790650