Footstep Planning for a Six-Legged in-Pipe Robot Moving in Spatially Curved Pipes

Sergei Savin, Ludmila Vorochaeva
Department of Mechanics, Mechatronics and Robotics
Southwest State University
Kursk, Russia
savinswsu@mail.ru

Abstract—This paper presents a footstep planning algorithm for a six-legged in-pipe robot moving in spatially curved pipes. The algorithm allows us to generate sequences of points on the pipe’s inner surface where the contact pads of the robot should come in contact with the surface. The algorithm uses mapping of the pipe onto a two-dimensional surface with a simpler geometry. The algorithm plans footstep sequences on that two-dimensional surface and then maps them back onto the pipe. It is shown that this procedure can be extended by implementing an obstacle avoidance algorithm formulated as a quadratic program.

Index Terms—In-pipe robot, footstep planning, spatially curved pipes, quadratic programming

I. INTRODUCTION

In-pipe mobile robots have been studied for the last three decades. The main tasks which these robots are designed to carry out include monitoring the state of pipelines, finding sections that need repair or maintenance work, cleaning pipes, mapping the existing pipelines and others [1-4]. These tasks require the robots to be able to move inside a pipeline carrying the specific equipment for the assigned task.

Many of the existing pipes are curved, have branches and varying diameter. The problem of efficiently moving through such pipes is still being investigated. One of the ways to partially solve this problem is to design robots that can traverse bends naturally with minimal intervention of the control system. Examples of such designs are shown in papers [5-7]. In publication [8] an approach to solve the problem of moving through a branched pipe is presented for robot composed of multiple sections. Here the control system plays a role by steering the robot into the right branch. In work [9] a robot with elastic elements designed to move in pipes with diameter varying along the length of the pipe is proposed.

Most of the robots described in the referenced above papers use wheels. Legged in-pipe robots can achieve a better mobility by their ability to adapt to the changing geometry of pipelines. Examples of legged in-pipe robots are shown in papers [10-12]. One of the important challenges for any legged robot is to plan a sequence of footsteps – a sequence of points where the robot is supposed to come in contact with the supporting surface. In paper [13] an algorithm for generating a sequence of footsteps is presented for a 9-link robot moving in a planar pipeline. The algorithm is able to handle curving pipes by producing step sequences with the spacing between the consecutive steps dependent on the local curvature of the pipe. The algorithm relies on a special geometric description of the pipeline that simplifies the footstep sequence generation. Here we consider a more general problem of planning footstep sequences for an in-pipe robot with a spatial structure moving in spatially curved pipes. We propose a geometric description of the pipe that will allow us to use simple footstep generation procedures.

We should note that although footstep planning for in-pipe robots still needs to be studied, there are a number of well-known studies on footstep planning for bipedal robots and other types of legged robots [14-16]. The results achieved in these studies include algorithms that take into account the geometry of the supporting surface and the obstacles present on it [15, 17]. Many of these results are applicable to the in-pipe robots. This is discussed in more detail in the following sections.

II. DESCRIPTION OF THE SIX-LEGGED IN-PIPE ROBOT

In this paper, we consider a walking type in-pipe robot with six legs. Each leg consists of two links connected via actuated rotational joints. The legs are attached to the robot’s body in such way that they form two symmetrical triplets, one behind the other. Each triplet assumes a T-shaped form when the legs are stretched out. Figure 1 shows a concept design of the robot.

It should be noted that the algorithms described in this paper are not exclusive for six-legged in-pipe robots and would work with a wide range of mechanical designs. The algorithms can be easily extended to handle different number of legs. It would also work for legs with a different structure compared to the one presented in fig. 1.
The links that are supposed to come in contact with the inner surface of the pipe have contact pads mounted on their tips.

Figure 2 shows a diagram of the robot.

In fig. 2 we use the following notation: $K_i$ are contact pads represented as a single point ($i = 1, 6$), $C_i$ and $D_i$ are actuated rotational joints, $C$ is the center of mass of the body link of the robot and $\psi_i$ and $\theta_i$ are angles that describe the relative orientation of the links that form robot’s legs.

III. GEOMETRIC DESCRIPTION OF THE PIPELINE

A. Centerline and Diameter Functions

The algorithms presented in this paper rely on a certain geometric description of the pipeline where the motion takes place. We consider pipes without branches and a circular cross section. This type of pipes can be described using the concept of centerline. A centerline is a curve whose points correspond to the centers of the cross sections of the pipe (each cross section is constructed by cutting the pipe with a plane perpendicular to the centerline). This is illustrated in fig. 3.

To completely describe a pipe we need to define its centerline curve $\xi$ as a function of a single parameter $s$ and define the diameter of the pipe $d$ as also a function of $s$:

$$\xi(s) = [\xi_x(s) \quad \xi_y(s) \quad \xi_z(s)]^T, \quad d = d(s),$$

where $\xi_x$, $\xi_y$ and $\xi_z$ are the Cartesian coordinates of the point on the centerline corresponding to a given value of $s$. The centerline function $\xi(s)$ must be differentiable. Although it is not necessary for the geometric description of the pipe, we additionally constrain the derivative of the function $\xi(s)$ with respect to $s$:

$$\left\| \frac{d\xi}{ds} \right\| = 1.$$  (2)

The constraint (2) allows us to interpret the parameter $s$ as a distance along the centerline and to connect it to the distance in Cartesian space. This constraint does not restrict the geometry of the centerline curve.

The position of each point on the pipe’s inner surface can be described by a pair of values $s$ and $\varphi$, where $\varphi \in [0 \quad 2\pi]$ is the angle determining the position of a point on the given cross section of the pipe.

B. Transformation to a Height Map and its Inverse

The chosen geometric description of the pipe allows us to define a transformation that maps a pipe to the so-called height map. The concept of height map is used in bipedal robotics to describe the supporting surface as a scalar function of two variables (see [18]). Here we define the height map as follows:

$$h(s, \varphi) = -d(s).$$  (3)
The height map function \( h(s, \phi) \) is a function of two scalar variables which can be thought of as local coordinates on the pipe’s surface. Here the function \( h(s, \phi) \) does not depend on \( \phi \) because the pipe has circular cross sections. For pipes with a more complex geometry \( h(s, \phi) \) might depend on \( \phi \).

The procedure of constructing the height map for a given pipe can be thought of as “unwrapping” the pipe. Figure 4 shows a pipe and its height map.

Since \( (s, \phi, h) \) systems to each point on the centerline. Let \( (s, \phi, h) \) form a coordinate system on a plane orthogonal to the centerline. This is the plane where the cross section of the pipe corresponding to \( s \) lies. We can then define a point \( p \) on the inner surface of the pipe as a function of \( s, \phi \) and \( h \):

\[
p(s, \phi, h) = \xi(s) - hT(e_1, \phi)e_2,
\]

where \( T(e_1, \phi) \) is a matrix representing the rotation by an angle \( \phi \) around axis \( e_1 \) and \( h \) is the height of the point \((s, \phi)\) on the height map defined by (3).

**IV. FOOTSTEP PLANNING**

The footstep planning algorithm must provide a sequence of states that define the position of contact points at the beginning of each step. For simplicity, we will only consider footstep planning for the forward leg triplet since the planning for the backward leg triplet is identical.

Let \( p_{A0}, p_{B0} \) and \( p_{C0} \) be initial positions of the contact points \( K_1, K_2 \) and \( K_3 \), and let \( k_{A0}, k_{B0} \) and \( k_{C0} \) be their images on the height map, represented by triplets of numbers \( s, \phi \) and \( h \):

\[
k_{A0} = [s_{A0}, \phi_{A0}, h_{A0}]^T,
\]

where \( s_{A0}, \phi_{A0} \) and \( h_{A0} \) are chosen such that if they are substituted into (4) the right hand side of the expression would be equal to \( p_{A0} \). The expressions for \( k_{B0} \) and \( k_{C0} \) are defined in a similar way.

Then we can define a step function that takes the position of a contact point before the step as an input and returns its position after the step as an output: \( k_{A,i+1} = f_{\text{step}}(k_{A,i}) \). This function can have the following form:

\[
k_{A,i+1} = \begin{bmatrix} s_{A,i} + \Delta s \\ \phi_{A,i} + \Delta \phi \\ h(s_{A,i} + \Delta s, \phi_{A,i} + \Delta \phi) \end{bmatrix},
\]

where \( \Delta s \) and \( \Delta \phi \) are constant displacements that define the step pattern of the robot and \( h(\cdot, \cdot) \) is the function (3). We can produce sequences of steps for contact points \( K_1, K_2 \) and \( K_3 \) using this step function as shown in fig. 6.
Then we can use the expression (4) to map these sequences from the height map to the actual pipe in three-dimensional space. This gives us sequences of the positions of points $K_1$, $K_2$, and $K_3$ that determine where the robot’s legs need to come into contact with the pipe’s inner surface. Figure 7 shows a pipe with contact points marked on it. The marks are placed on both the inner and on the outer surfaces of the pipe for the clarity.

Analyzing the results of the operation of the step generation algorithm, we can note that it produces very simple sequences of steps on the height map. These sequences are determined by the desired step function (6) and can be controlled by manipulating parameters $s$ and $\Delta \varphi$. However, the mapping of these sequences onto the pipe demonstrates a more varied behavior. The sequences change their spacing (the distance between the neighboring points) depending on the geometry of the pipe. In particular, when a sequence goes through an L-bend or a U-bend it would become more spaced out if it is going through the side of the pipe that is located further from the axis of the bend. This avoids the problem of one sequence “outrunning” the others. This behavior is the natural consequence of generating the sequences on the height map and then using expression (4) to map them back onto the pipe. A similar effect is observed in [13] where a similar footstep generation algorithm for planar pipelines is discussed.

V. OBSTACLE AVOIDANCE ALGORITHM

Another advantage of using the height map representation of the pipeline is the possibility to adapt obstacle avoidance algorithms designed for bipedal walking robots for use with the in-pipe robots. Examples of obstacles inside pipelines include deposits of soft matter that will not be able to provide an adequate support for the robot’s legs, pipe branches that can be viewed as holes in the supporting surface, and other types of structural elements that significantly change the geometry of the supporting surface. The footstep planning algorithm described in this paper is designed to work with the pipes with circular cross sections, so in practice any region of the pipe’s inner surface that deviates from it may be considered an obstacle.

Let us assume that the obstacles on the inner surface of the pipe are represented by their vertices (in practice we can use bounding polygons or polyhedra to approximate the obstacles whose shapes that cannot be represented by a set of vertices). Then we can map these vertices onto the height map. Figure 8 shows an example of such mapping.

Then we can use the algorithms presented in [19-20] to generate obstacle-free convex regions on the height map, represented by systems of linear inequalities. With this we can produce a correction procedure for every foot placement position. The procedure is implemented as the following quadratic program:

$$
\min \mathbf{e}^T \mathbf{W} \mathbf{e}
$$

subject to

$$
\mathbf{A} j \begin{bmatrix} \mathbf{s}_j \\ \varphi_j \end{bmatrix} + \mathbf{e} < \mathbf{b}_j,
$$

(7)
where $\mathbf{e}$ is a displacement vector that defines where relative to the original position should the contact point be placed, $\mathbf{A}_j$ and $\mathbf{b}_j$ are a matrix and a vector that correspond to a linear inequality representation of a convex obstacle-free region, and $\mathbf{W}$ is a positive-definite weight matrix. The value of $\mathbf{W}$ can be chosen such that it penalizes sideways displacements more.

The problem (7) is solved for each obstacle free region and then the solution with the minimal value of the cost function is chosen. Figure 9 illustrates the results of the algorithm’s operation.

![Fig. 9. Height map with footstep sequences marked on it (top view); 1 – the surface defined by function (3), 2 – the opening in the pipe, 3 – markers that indicate the originally generated footstep positions; 4 – markers that indicate footstep positions after the correction](image)

The solution shown in fig. 9 was obtained for the following value of $\mathbf{W}$:

$$
\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}.
$$

(8)

This choice of $\mathbf{W}$ gives the algorithm an incentive to shift the foot placement by changing its $\varphi$ coordinate rather than $s$.

We can observe that the sequence of footstep positions obtained after the correction procedure avoids the obstacle. After the correction is applied, the corrected sequence is mapped back onto the pipe, using formula (4). The result is shown in fig. 10.

![Fig. 10. Pipe with footstep sequences marked on it; 1 – the pipe, 2 – the opening in the pipe 3 – markers that indicate originally generated footstep positions; 4 – markers that indicate footstep positions after the correction](image)

VI. CONCLUSIONS

In this paper, a six-legged in-pipe walking robot was considered. The focus of the paper was the discussion of a footstep planning procedure that allows the robot to move in spatially curved pipes. The presented algorithm relies on a particular geometric description of the pipe where the motion takes place. This geometric description allows mapping of the pipe to a height map with a much simpler geometry. Then a simple procedure can be used to generate the footstep sequences on the height map and map them back onto the pipe. This allows us to generate feasible footstep sequences on spatially curved pipes using relatively simple computations. This is advantageous because in practice the algorithm has to be implemented on on-board computers of in-pipe robots that might have low computational power.

Additionally, it was shown the proposed step generation procedure can easily be extended to implement a simple optimization-based obstacle avoidance algorithm. The obstacle avoidance algorithm uses quadratic programming, which can be solved in real time [22].

One of the questions not discussed in the paper is localization of the robot in the pipe. The presented footstep generation algorithm requires the robot to know its position relative to the pipe, which can not be measured directly. This topic will be considered as part of future work on this project.

ACKNOWLEDGMENT

This work is supported by the Presidential grant MK-2577.2017.8.

REFERENCES


Jatsun, S., Savin, S. and Yatsun, A., 2016, October. Study of controlled motion of an exoskeleton performing obstacle avoidance during a single support walking phase. In System Theory, Control and Computing (ICSTCC), 2016 20th International Conference on (pp. 113-118). IEEE.