

# Trajectory generation for a walking in-pipe robot moving through spatially curved pipes

Sergei Savin\*, Sergey Jatsun and Lyudmila Vorochaeva

Southwest State University, Department of Mechanics, Mechatronics and Robotics, 305040 Kursk, Russia

**Abstract.** In this paper, a walking in-pipe robot is studied. The robot has six legs, each consisting of 3 links connected via rotary joints. The legs are attached to the robot's body. The work is focused on the problem of generating desired position and orientation for the robot's body, using a given footstep sequence. An iterative geometric algorithm for generating orientation sequence is proposed. The problem of finding the desired position of the center of mass of the robot's body is formulated as a problem of minimizing stretching of the robot's legs during steps. Also, an analytical solution for inverse kinematics problem has been given. All proposed algorithms do not require extensive calculation and use basic algebraic operations.

## 1 Introduction

The main use of in-pipe robots is the transport of specialized equipment, which, in its turn, is used for diagnostics and maintenance of the pipeline [1, 2]. Having the equipment inside the pipeline is sometimes the only option other than excavation of the pipe, which is associated with significant costs. Additionally, being able to access the inner surface of the pipe can provide for a better detection of defects that occur there. It also provides a direct way to monitor the deposits on the inner surface of the pipe. Sensor systems employed in such inspection tasks are discussed in paper [3]. The use of in-pipe robots for cleaning the pipelines was also proposed in [4].

There are several types of in-pipe robots. Many of the proposed classifications distinguish seven or more different types [5, 6]. Paper [7] is devoted to the study of motion of worm-like robot; the work [8] describes the parallel-link robot with flexible structure. Papers [9,10] present the results of numerical simulations in-pipe robot based on the vibration principle of movement. All of the above robots are used for inspection of pipelines. In this paper we focus on walking in-pipe robots. This type of robots is particularly interesting because of its ability to "step over" obstacles, potentially allowing such robots to navigate pipelines with branches, matter deposits and other types of obstacles.

Some of the walking robot designs are shown in papers [11-17]. In works [11-13] an eight legged in-pipe robot and its sensory system are studied. In paper [14] a six legged in-pipe robot designed for motion in spatially curved pipes is discussed. Articles [15-17] focus on planar in-pipe robots with four legs, considering designs with a solid and segmented main link.

We should note that there are a number of problems associated with controlling walking in-pipe robots. That

includes step sequence generation, trajectory generation, inverse kinematics, controller design that takes into account information about contact with the inner surface of the pipe, observer design and others. This is similar to the set of problems associated with other types of walking robots. Many of the solutions for these problems found for different types of walking robots are viable for in-pipe walkers as well; this is discussed in paper [14] in more detail.

Some of the listed above problems have been studied specifically for the case of in-pipe walking robots. In papers [14,15] step sequence generation algorithms are proposed for planar and spatial in-pipe walking robots. In work [15] an analytic solution for inverse kinematics problem is given for a 9 link planar robot. In [16] a quadratic programming-based controller design for an in-pipe walking robot is proposed, able to take into account actuator torque limits and mechanical constraints, preventing loss of contact and slipping of the leg. Additionally, inverse kinematics algorithms, control designs and state-estimator designs were studied in [18] for a bipedal walking robot, using approaches that can be translated to in-pipe robots.

In this paper we focus on trajectory generation problem for the robot's body (the main link that all robot's legs are connected to). Knowing position and orientation of robot's body and robot's footstep sequence allows us to solve inverse kinematics problem for each leg individually, obtaining all joint angles. If there are three joints in each leg and the end effector of the leg (its foot) has three translational degrees of freedom when the robot's body is fixed, then in this case the inverse kinematics problem has a finite number of solutions that may be found analytically. For such a case, it should be possible to write an efficient inverse kinematics algorithm that can be executed using on-board computer without producing excessive computational load. Also,

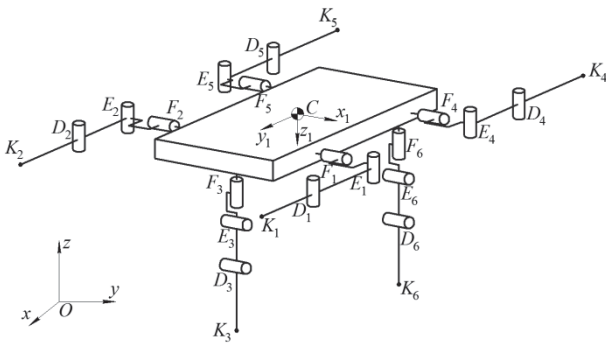
\* Corresponding author: [savinswsu@mail.ru](mailto:savinswsu@mail.ru)

knowing robot's body position and orientation allows us to estimate the feasibility of the robot's trajectory with respect to the geometry of the pipeline.

This paper can be considered a follow-up on the work [15], where the problem of body position and orientation generation was solved for a planar robot. The goal of the paper is to derive a systematic approach for generating center of mass trajectory for the robot's main link, as well as time functions defining its orientation.

## 2 Robot description

In this paper, we consider a six legged in-pipe robot with 19 individual links. Each leg includes three links connected via rotary joints and is attached to the robot's body. The construction of the robot is shown in Figure 1.



**Fig. 1.** Construction of the six-legged walking in-pipe robot: 1 – robot's body, 2 – robot's legs.

In Figure 1 point  $C$  denotes the center of mass of the body link,  $F_i$ ,  $E_i$  and  $D_i$  denote robot's joints and points  $K_i$  denote contact pads on the end of the robot's legs,  $i = \overline{1,6}$ . We denote the position of these points as  $\mathbf{r}_C$ ,  $\mathbf{r}_{F_i}$ ,  $\mathbf{r}_{E_i}$ ,  $\mathbf{r}_{D_i}$  and  $\mathbf{r}_{K_i}$  accordingly. The orientation of the body is defined by a directional cosines matrix  $\mathbf{T}$ .

If  $\mathbf{r}_C$  and  $\mathbf{T}$  are given, then we can determine  $\mathbf{r}_{F_i}$  using the following expression:

$$\mathbf{r}_{F_i}^{(0)} = \mathbf{T}\mathbf{r}_{F_i}^{(1)} + \mathbf{r}_C^{(0)}, \quad (1)$$

where the superscript denotes the reference frame,  $\mathbf{r}^{(0)}$  means the vector is expressed in world frame  $Oxyz$  and  $\mathbf{r}^{(1)}$  means it is expressed in local frame  $Cx_1y_1z_1$ .

Furthermore, we assume that links  $F_iE_i$  have such shape, that rotations in joints  $F_i$  don't change the positions of points  $E_i$ .

## 3 Determining desired position and orientation of the robot's body

The algorithm described in this section discretizes robot's trajectory and aims to find  $\mathbf{T}$  and  $\mathbf{r}_C$  for every discrete point in time  $t_j$ . We assume that we are given vectors  $\mathbf{r}_{K_i}(t_j)$  for the current time point  $t_j$ , and we are given matrix  $\mathbf{T}(t_{j-1})$  and vector  $\mathbf{r}_C(t_{j-1})$  for the

previous time point  $t_{j-1}$ . The algorithm for generating footstep sequence  $\mathbf{r}_{K_i}(t_j)$  is described in detail in [14].

The algorithms includes two steps. First, we find the desired orientation of the robot's body, and then the position of its center of mass.

### 3.1. Determining desired orientation

First, we define directional vector  $\sigma_K$  for the footsteps  $\mathbf{r}_{K_i}$  as follows:

$$\sigma_{K,i} = \frac{\mathbf{r}_{K,i+3} - \mathbf{r}_{K,i}}{\|\mathbf{r}_{K,i+3} - \mathbf{r}_{K,i}\|}, \quad (2)$$

$$\sigma_K = \frac{\sigma_{K,1} + \sigma_{K,2} + \sigma_{K,3}}{\|\sigma_{K,1} + \sigma_{K,2} + \sigma_{K,3}\|}, \quad (3)$$

Then, using  $\mathbf{T}(t_{j-1})$  and formula (1) we can find  $\mathbf{r}_{F_i}(t_{j-1})$ , the positions of the body joints during the previous step. We define directional vector  $\sigma_F$  for robot's body the same way as we defined  $\sigma_K$ , only instead of  $\mathbf{r}_{K_i}(t_j)$  we use  $\mathbf{r}_{F_i}(t_{j-1})$ .

Finally we can find rotation matrix that takes  $\sigma_F$  to  $\sigma_K$ . To do this we find axis of rotation  $\mathbf{e}_{FK}$  and angle of rotation  $\alpha_{FK}$ :

$$\mathbf{e}_{FK} = \frac{\sigma_F \times \sigma_K}{\|\sigma_F \times \sigma_K\|}, \quad (4)$$

$$\alpha_{FK} = \cos^{-1}(\sigma_F \cdot \sigma_K). \quad (5)$$

The directional cosines matrix  $\mathbf{T}(t_j)$  is then given by the following expression:

$$\mathbf{T}(t_j) = \mathbf{T}(\mathbf{e}_{FK}, \alpha_{FK})\mathbf{T}(t_{j-1}), \quad (6)$$

where  $\mathbf{T}(\mathbf{e}_{FK}, \alpha_{FK})$  is the matrix representing rotation around axis  $\mathbf{e}_{FK}$  by angle  $\alpha_{FK}$  [19].

We should note that this method can accumulate error associated with roll angle (in roll-pitch-yaw representation of rotations). This can be avoided by performing an additional rotation, using  $\sigma_K$  as an axis.

### 3.2. Determining desired position

To find desired position of the center of mass  $\mathbf{r}_C(t_j)$  we find the vector  $\mathbf{p}$  that shifts the previous its position  $\mathbf{r}_C(t_{j-1})$  to the new one, such that the resulting new positions of points  $F_i$  minimize the distance between them and points  $K_i$ . This condition aims at minimizing stretching of the robot's legs, allowing them to stay further away from kinematic singularities.

Then the vector  $\mathbf{p}$  given by the following expression:

$$\mathbf{p} = \operatorname{argmin}_{\mathbf{p}} \sum_{i=1}^6 \left\| \mathbf{T}(t_j) \mathbf{r}_{Fi}^{(1)} + \mathbf{r}_C^{(0)}(t_{j-1}) + \mathbf{p} - \mathbf{r}_{Ki}^{(0)} \right\|. \quad (7)$$

This can be formulated as a quadratic program or can be solved directly using least squares method. To do the latter, we define a difference vector  $\mathbf{d}$  and matrix  $\mathbf{C}$  :

$$\mathbf{d} = \begin{bmatrix} \mathbf{T}(t_j) \mathbf{r}_{F1}^{(1)} + \mathbf{r}_C^{(0)}(t_{j-1}) - \mathbf{r}_{K1}^{(0)} \\ \vdots \\ \mathbf{T}(t_j) \mathbf{r}_{F6}^{(1)} + \mathbf{r}_C^{(0)}(t_{j-1}) - \mathbf{r}_{K6}^{(0)} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix}, \quad (8)$$

where  $\mathbf{I}$  is a 3 by 3 identity matrix, and both  $\mathbf{C}$  and  $\mathbf{d}$  have 18 rows. Then the solution for least squares can be found as:

$$\mathbf{p} = \mathbf{C}^+ \mathbf{d} = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{d}. \quad (9)$$

Thus, the new position of the center of mass is given as:

$$\mathbf{r}_C(t_j) = \mathbf{r}_C(t_{j-1}) + \mathbf{p}. \quad (10)$$

### 3.3. Generating continuous trajectories

In order to generate continuous trajectories from the obtained sequence of positions and orientations, we can use interpolation. Sequence  $\mathbf{r}_C(t_j)$  can be directly interpolated using splines. In order to interpolate orientations, they need to be converted to a different representation, because interpolating sequence of directional cosines matrix can produce matrices that would not be orthonormal and thus would require orthogonalization. Instead, we can replace matrices  $\mathbf{T}(t_j)$  with quaternion  $\mathbf{q}(t_j)$  describing equivalent rotations [19]. This sequence of quaternions can then be interpolated using special algorithms, such as Slerp [20]. This interpolation can be used to recover rotation matrices  $\mathbf{T}$  for any moment in time, or directly, by modifying formula (1) and other kinematics expressions accordingly.

Figures 2 and 3 shows a sequence of positions of robot's body inside a pipeline (legs are not drawn, pipeline is opaque in Figure 2 and transparent in Figure 3). Figure 2 shows that the proposed method allows generating a sequence of positions of the center of mass of the robot. The robot always stays roughly at the center of the pipe, and keeps its orientation "parallel to the pipeline's direction".

## 4 Inverse kinematics algorithm

Having  $\mathbf{T}(t)$  and  $\mathbf{r}_C(t)$  we can find  $\mathbf{r}_{Fi}(t)$  using (1). Furthermore, we can find  $\mathbf{r}_{Ei}(t)$  with the next formula:

$$\mathbf{r}_{Ei}^{(0)} = \mathbf{T} \mathbf{r}_{FiEi}^{(1)} + \mathbf{r}_{Fi}^{(0)}, \quad (11)$$

where  $\mathbf{r}_{FiEi}$  is vector drawn from point  $F_i$  to  $E_i$ . Figure 4 shows the leg of the robot. In the Figure 4,  $l_1$  and  $l_2$  and lengths of the links  $E_i D_i$  and  $D_i K_i$  and  $\psi_i$ ,  $\phi_{i,1}$  and  $\phi_{i,2}$  are joint angles.

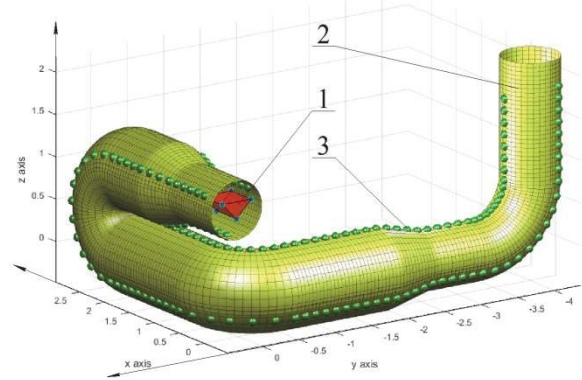


Fig. 2. Spatially curved pipeline with robot in it; 1 – initial robot's position, 2 – pipeline, 3 – points where the robot steps to.

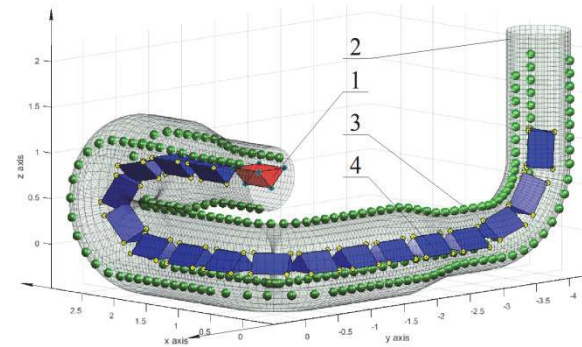


Fig. 3. Sequence of positions for robot's body inside a pipeline; 1 – initial robot's position, 2 – pipeline, 3 – points where the robot steps to, 4 – intermediate positions.

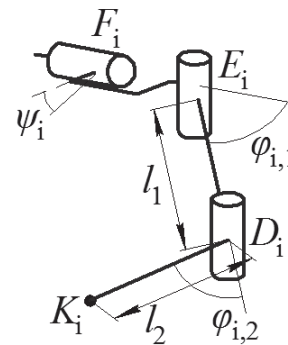


Fig. 4. Construction of the robot's leg.

Let us consider vector  $\mathbf{r}_{EiKi}$  drawn between points  $E_i$  and  $K_i$ . Using formula (11) and the given sequence  $\mathbf{r}_{Ki}(t_j)$  (or its continuous version) we can find the value of this vector. Then we can calculate the length of this vector  $L_i$  and the azimuth angle  $\psi_i$  in the following way:

$$L_i = \|\mathbf{r}_{E_iK_i}\|, \quad \psi_i = \tan^{-1} \frac{y_1 \cdot \mathbf{r}_{E_iK_i}}{z_1 \cdot \mathbf{r}_{E_iK_i}}, \quad (12)$$

Then we rotate the vector  $\mathbf{r}_{E_iK_i}$  so it would lie parallel to  $Oz_x$  plane :

$$\mathbf{r}_{E_iK_i}^* = \mathbf{T}_x(\psi) \mathbf{T}^{-1} \mathbf{r}_{E_iK_i}, \quad (13)$$

With this we can consider a simple planar inverse kinematics problem, illustrated in Figure 5.

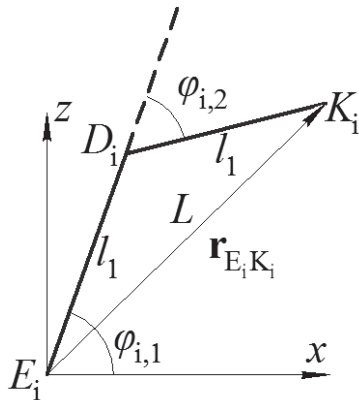


Fig. 5. Inverse kinematics diagram.

The solution for the planar inverse kinematics problem is given by the next expressions:

$$\varphi_{i,1} = \tan^{-1} \left( \frac{z_{E_iK_i}^*}{x_{E_iK_i}^*} \right) + \cos^{-1} \left( \frac{l_1^2 + L^2 - l_2^2}{2l_1L} \right), \quad (14)$$

$$\varphi_{i,2} = \pi - \cos^{-1} \left( \frac{l_1^2 + l_2^2 - L^2}{2l_1l_2} \right), \quad (15)$$

Using these expressions we can solve for  $\psi_i$ ,  $\varphi_{i,1}$  and  $\varphi_{i,2}$  triplets for each leg, thus solving inverse kinematics problem for the robot. Figure 6 shows the result of applying these algorithms, demonstrating obtained orientations of robot's legs while it moves through a curved pipe.

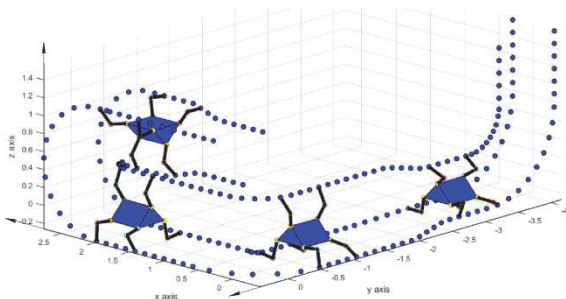


Fig. 6. Robot with legs drawn, their orientation found by the inverse kinematics algorithm (the pipe is not shown).

We can see that the robot bends its “knees” in the same direction, which is the result of the chosen approach. This can be controlled by changing signs in

the expressions (14) and (15). Similar problem was discussed in [15, 21].

We should note that this algorithm does not handle singularities, and it would fail in the case when a point can't be reached. There are easy ways to check if the algorithm will fail, for example by using the following condition:

$$L_i \leq l_1 + l_2. \quad (16)$$

If the leg segments are not of equal length, we can add another condition, requiring that the length of the longer segment would be less or equal to the length of the other segment plus  $L_i$ . If the conditions are not satisfied, then the inverse kinematics problem for the leg has no solution.

## 6 Conclusions

In this paper a six legged in-pipe walking robot was studied. The aim of the study was to provide an efficient algorithm for generating positions and orientations of the robot's body, and to show how it can be used to derive an analytic solution for inverse kinematics problem.

The proposed method for generating center of mass trajectory is based on interpolation of the sequence of positions of the center of mass. That sequence is in its turn found by considering the problem of minimizing the distance between the footsteps locations (points of contact where the robot needs to step to) and the bases of robot's legs. This problem can be solved algebraically for the case when robot's orientation is fixed.

The orientation is found by a heuristic-based approach, which uses so-called directional vectors, which indicate where the robot should be orientated. The procedure of generating a sequence of orientations is iterative and requires initialization, although the initial orientation supplied to the algorithm does not need to be very accurate. Then the sequence of orientations can be represented as a sequence of quaternions and interpolated.

The proposed inverse kinematics solution is based on simple geometric manipulations. The advantage of this approach, same as with the other methods proposed in this paper, is its low computational requirements.

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

## References

1. H.T. Roman, B.A. Pellegrino, W.R. Sigrist, *IEEE Transactions on Energy Conversion*, **8**(3), 576-583 (1993)
2. J.M. Mirats Tur and W. Garthwaite, *Journal of Field Robotics*, **27**(4), 491-508 (2010)
3. O. Duran, K. Althoefer, L.D. Seneviratne, *IEEE Sensors journal*, **2**(2), 73-81 (2002)
4. C.D. Jung, W.J. Chung, J.S. Ahn, M.S. Kim, G.S. Shin, S.J. Kwon, *International Conference on*

- Mechatronics and Automation (ICMA)*, 1327-1332 (2011)
5. N.S. Roslin, A. Anuar, M.F.A. Jalal and K.S.M . Sahari, *Procedia Engineering*, **41**, 1456-1462 (2012)
  6. S.G. Roh, H.R. Choi, *IEEE transactions on robotics*, **21**(1), 1-17 (2005)
  7. Jatsun S., Malchikov A., *Handbook of Research on Advancements in Robotics and Mechatronics*, 168-218 (2014)
  8. S. Jatsun, A. Yatsun, S. Savin. *15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2012*, 713-719 (2012)
  9. G. Carbone, M. Ceccarelli, S. Jatsun, A. Malchikov, *10th IFToMM International Symposium on Science of Mechanisms and Machines*, 103-114 (2010)
  10. A.N. Grankin, S.F. Yatsun. *Journal of Computer and Systems Sciences International*, **48**(1), 155-163 (2009)
  11. J.A. Gálvez, P.G. De Santos, F. Pfeiffer, *IEEE/ASME Transactions on mechatronics*, **6**(1), 26-35 (2001)
  12. A. Zagler, F. Pfeiffer, *IEEE International Conference on Robotics and Automation ICRA'03*, 2954-2959 (2003)
  13. F. Pfeiffer, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, **365**, 109-131 (2007)
  14. S. Savin, L.Yu. Vorochoeva, *13th Siberian conference SIBCON-2017* (to be published)
  15. S. Savin, L. Vorochoeva, *International Conference on Industrial Engineering 2017*, (to be published)
  16. S. Savin and L.Yu. Vorochoeva, *International Conference On Industrial Engineering 2017*, (to be published)
  17. S. Jatsun, A. Malchikov, O. Loktionova, *XX CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators ROMANSY 2014*, 341-348 (2014)
  18. S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, R. Tedrake, *Autonomous Robots*, **40**(3), 429-455 (2016)
  19. J. Diebel. *Matrix*, **58**(15-16), 1-35 (2006)
  20. K. Shoemake. *ACM SIGGRAPH computer graphics* **19**(3), 245-254 (1985)
  21. L.Yu. Vorochoeva, A.S. Yatsun, S.F. Jatsun. *SPIIRAS Proceedings*. **3**(52), 70-94 (2017)