

State observer design for a walking in-pipe robot

Sergei Savin*, Sergey Jatsun and Ludmila Vorochaeva

Southwest State University, department of mechanics, mechatronics and robotics, 305040 Kursk, Russia

Abstract. In this paper, a state observer design for a walking in-pipe robot is studied. The necessity of using a state observer is related to the fact that sensors have limited accuracy and are prone to producing noise. This is especially problematic for in-pipe walking robots, since they use model-based control and require accurate information of their current state. The paper shows that an iterative state observer based on solving Riccati equation provides significant improvements in the behaviour of the control system. It allows to smooth out the spikes in the control actions requested by controller and to minimize tremor of the robot links. In order to study the behaviour of the observer when different sensors are used, a performance function was introduced. It was shown that the observer allows to improve the performance of the control system for a wide range of sensor parameters. Additionally, it was shown that the introduction of the observer allows to choose higher feedback controller gains, enabling more precise control. Simulations on the full robot model, taking into account mechanical constraints and contact forces showed that the linear observer is capable of improving the behaviour of the control system of the walking robot, if measurements of the reaction forces are provided. The effects that the noise and quantization in the reaction forces measurements have on the behaviour of the state observer is studied.

1 Introduction

Pipeline inspection robots are complex electro-mechanical systems that are designed to move inside pipelines while carrying necessary equipment. Popular in-pipe robot designs feature parallel linkages with wheels or tracks [1-4]. Some other designs include flexible links, pneumatic or Bowden cable activation, serial linkages designs (such as snake-like robots), multi-module designs, etc. [5-7]. This diversity and complexity is the result of the diversity and complexity of the tasks posed for this type of robots and reflects the advances in the field of in-pipe robotics.

Currently, in-pipe robot designs have been able to solve a number of mobility tasks, however the problem of navigating pipelines has not been completely solved yet. In part, this is due to the diversity of the mobility tasks the in-pipe robots face. For example, pipelines can have significant changes in diameter, vertical sections, L-shaped and U-shaped turns, Y-shaped and T-shaped junctions and other features that make it difficult to navigate for most of the known robot designs. This motivates the use of more agile robot designs, such as walking in-pipe robots [8-10].

All types of in-pipe robots require solving a number of sensor-related challenges. These include estimation of the robot's position and configuration, navigation and mapping. Some of the sensor-related problems are common for all types of multilink robots, and arise due to the fact that real sensors are characterized by quantization, discretization, noise, time lag and other strongly nonlinear behavioural features. Estimating the

configuration of walking in-pipe robots is an additionally challenging problem because of their structure, which features multiple serial linkages connected to the floating body (for the terminology see [11]). Since calculating the positions of links at the end of a sequential kinematic chain requires measurements of a number of joint parameters (angles or linear displacements), the nonlinear behaviour of the joint sensors will manifest significantly in these calculations. Incidentally, the contact elements of walking in-pipe robots, whose trajectories need to be controlled [12-15], are located at the ends of sequential kinematic chains and therefore their positions are hard to infer exactly. This motivates the use of special configuration estimation algorithms.

There are a number of methods that can be used for configuration estimation (or state estimation). For linear systems, state observers can be used. In papers [16-18] state observers had been used to control DC motors, underwater robots and wheeled mobile robots. This shows that although these methods are developed for linear systems, they can be used for nonlinear mechanical systems as well. This is similar to how linear control methods, such as linear quadratic regulator, can be used to control robots with nonlinear dynamics (see [19-21] for examples). In this paper, we look at the use of linear state observer to provide predictions for joint parameters of an in-pipe robot.

The rest of the paper is organized as follows. The section 2 provides description on an in-pipe walking robot which is used in the numerical studies. The section 3 discusses the observer design, and sections 4 and 5 show the work of the observer in simulation.

* Corresponding author: savinswsu@mail.ru

2 Walking in-pipe robot description

In this paper, we consider an in-pipe walking robot with nine links. One of the links is called "body" and other links are organized into pairs and are connected to the body in a serial manner. The robot is shown in figure 1.

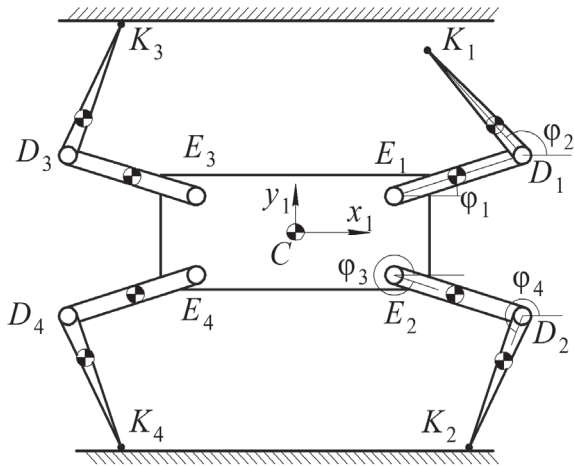


Fig. 1. Diagram of the in-pipe walking robot.

The diagram in fig. 1 uses the following notation. D_i and E_i are active joints, K_i are contact elements, φ_i are joint angles, x_1 and y_1 are axes of a local coordinate system, attached to the robot's body and C is the center of mass of the robot's body.

In the following sections, we will consider one of the robots legs separately from the rest of the device. Assuming that the robot's body remains stationary, we can introduce a vector of generalized coordinates that describes the configuration of one of the robot's legs:

$$\mathbf{q} = [\varphi_1 \quad \varphi_2]^T. \tag{1}$$

Using this generalized coordinates we can describe robot's dynamics in the following way:

$$\mathbf{H}\dot{\mathbf{q}} + \mathbf{c}_0 = \boldsymbol{\tau}, \tag{2}$$

where \mathbf{H} is the generalized inertia matrix, \mathbf{c}_0 is a bias vector and $\boldsymbol{\tau}$ is a generalized motor torques (see [22] for terminology, derivations and practical algorithms for generating these equations). These equations can be linearized in the following way:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{c}, \tag{3}$$

where $\mathbf{x} = [\mathbf{q}^T \quad \dot{\mathbf{q}}^T]^T$ is a vector of state coordinates, $\mathbf{u} = \boldsymbol{\tau}$, $\mathbf{A} = \partial \mathbf{f} / \partial \mathbf{x}$, $\mathbf{B} = \partial \mathbf{f} / \partial \mathbf{u}$ and $\mathbf{c} = \mathbf{f} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}$ are constant linearization matrices and vectors calculated at a given point \mathbf{x}_0 and $\mathbf{f} = \mathbf{H}^{-1}(\boldsymbol{\tau} - \mathbf{c}_0)$ is the function describing the dynamics of the robot.

The robot is controlled by a computed torque controller (CTC), which is given by the following control law:

$$\boldsymbol{\tau} = \mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{c}_0, \tag{4}$$

where $\ddot{\mathbf{q}}^*$ is the desired value of the generalized coordinates, $\mathbf{e} = \mathbf{q}^* - \mathbf{q}$ is the control error and \mathbf{K}_p and \mathbf{K}_d are gain matrices. The use of this controller is discussed in [23] and the tuning of the gain matrices is studied in [24-25].

3 State observer design

The idea behind a state observer is to predict the actual evolution of the state coordinates of the robot given a model of the robot dynamics and possibly incorrect measurements of the state coordinates. Let us introduce a vector of state coordinate predictions \mathbf{z} . We define the observer dynamics as follows:

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} + \mathbf{c} + \mathbf{L}(\mathbf{x} - \mathbf{z}), \tag{5}$$

where \mathbf{L} is a gain matrix of the observer. In order to find the optimal gain matrix we can define $\mathbf{e} = \mathbf{x} - \mathbf{z}$, subtract (5) from (3) and simplify to get:

$$\begin{cases} \dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{I}\bar{\mathbf{u}} \\ \bar{\mathbf{u}} = -\mathbf{L}\mathbf{e} \end{cases}, \tag{6}$$

where \mathbf{I} is an identity matrix. This suggests that the optimal gain matrix \mathbf{L} can be computed by solving the Riccati equation, same as it is done in linear quadratic regulator design. Examples of how this is done can be found in [20-21]. The weight matrices that appear in the Riccati equations are chosen to be identity matrices in all experiments shown in this paper.

If the mechanical system has explicit mechanical constraints imposed on it, the observer dynamics equation can be modified as follows:

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} + \mathbf{F}^T \boldsymbol{\lambda} + \mathbf{c} + \mathbf{L}(\mathbf{x} - \mathbf{z}), \tag{7}$$

where \mathbf{F} is the constraint Jacobian taken with respect to the state coordinates \mathbf{x} and $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers that are equivalent to reaction forces. Values of $\boldsymbol{\lambda}$ can be read from force sensors, and they are subject to measurement errors.

Thus, the observer is given by the update law (5) or (7), where the gain matrix is obtained by solving the Riccati equation for the dynamical system (6). For nonlinear system this is done iteratively, by linearizing the system along the path that the system follows in the state space. In order to prevent the erroneous values of sensor measurements from appearing in the dynamics model during linearization, the model can be linearized along its desired (rather than actual) trajectory in the state space. This is what is done in all experiments shown in this paper.

4 Simulation results

In this section, we study the behaviour of a single leg of the walking in-pipe robot, taking into account the nonlinear effects of the joint sensors. In particular, we

model sensor quantization and sensor noise. The sensor model is then given as follows:

$$\tilde{x} = x - (x \bmod \sigma) + r(c, \delta), \quad (8)$$

where \tilde{x} is the measured value of x , $(x \bmod \sigma)$ is remainder after division of x by σ and $r(c, \delta)$ is a random number that has a flat distribution in the interval $[c - 0.5\delta \quad c + 0.5\delta]$. Figure 2 shows the sensor output during a single robot step for the case when $c = 0$, $\delta = 0.05$ and $\sigma = 0.05$ for both position and velocity sensors.

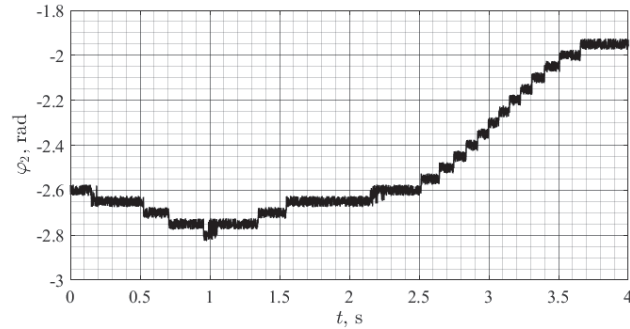


Fig. 2 The measured values of $\varphi_2(t)$.

As we can see, the measured value of joint angle increases by visible increments, and the noise levels are so high that they make it difficult to determine the time when the increments took place. Expression (4) suggests that the nonlinearities in $\varphi(t)$ translate to nonlinearities in $\tau(t)$. Figure 3 shows the generalized torques $\tau(t)$ obtained for the case discussed above. This and the following simulations are done with $\mathbf{K}_p = 1000\mathbf{I}$ and $\mathbf{K}_d = 100\mathbf{I}$, where \mathbf{I} is a 2-by-2 identity matrix.

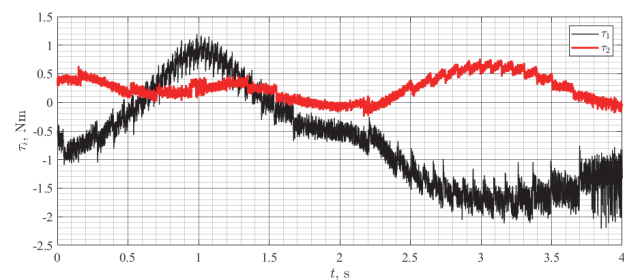


Fig. 3. The obtained values of $\tau_1(t)$ and $\tau_2(t)$ for the experiment with sensor noise and quantization.

Figure 3 shows that the torques change rapidly, possibly faster than is possible for some motor types. This illustrates another reason why state estimation is important, since it is needed to prevent the control system from requesting impossible behaviour from the robot motors.

Let us study how the behaviour of the controller changes when the state observer (5) is introduced. Figure 4 shows the time functions $\tau(t)$ for this case.

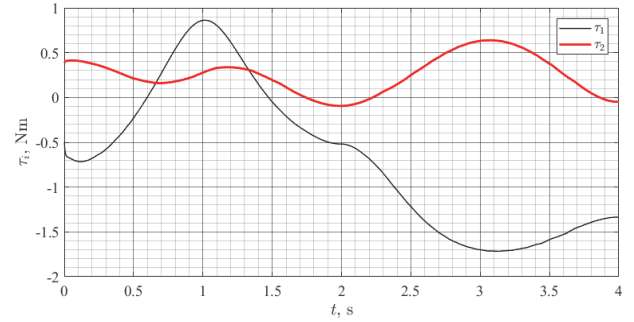


Fig. 4. The obtained values of $\tau_1(t)$ and $\tau_2(t)$ for the experiment with sensor noise and quantization, with the state observer used.

As we can see, for this experiment the time functions $\tau(t)$ requested by the controller are smooth, which suggests that the state observer works well. To make this observation more quantitative, let us introduce a cost function J :

$$J = \int [(\mathbf{x}^* - \mathbf{x})^T \mathbf{Q}(\mathbf{x}^* - \mathbf{x}) + (\mathbf{u}^* - \mathbf{u})^T \mathbf{R}(\mathbf{u}^* - \mathbf{u})] dt, \quad (9)$$

where \mathbf{x}^* is the desired value of \mathbf{x} and \mathbf{u}^* is the value of \mathbf{u} found using inverse dynamics and \mathbf{Q} and \mathbf{R} are weight matrices of the additive quadratic cost J . In the following experiments we use the weights $\mathbf{Q} = 1$ and $\mathbf{R} = 0.1$.

Let us study the cost J as a function of the sensor resolution σ . Figure 5 shows this graph for the case when observer is not used.

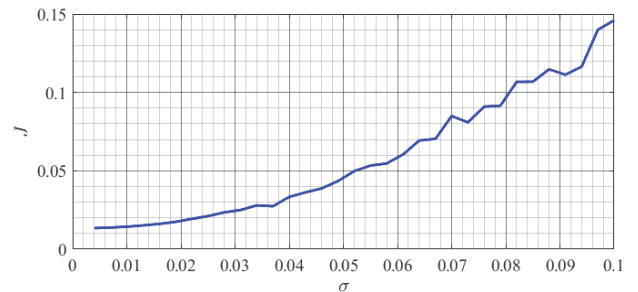


Fig. 5. Dependence of J on sensor resolution σ (without using the observer, $c=0$, $\delta=0.05$)

As we can see, the cost function steadily grows as σ increases (the sensor resolution falls). This can be expected from the shape of the graphs shown in Figure 3. The same dependence for the case when the linear observer is used is shown in Figure 6.

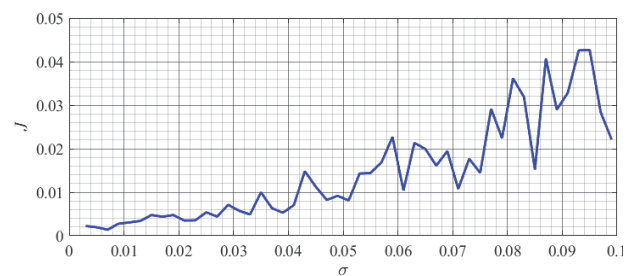


Fig. 6. Dependence of J on sensor resolution σ (with using the linear observer, $c=0$, $\delta=0.05$).

Figure 6 shows that introducing the state observer improves the cost. When the state observer is used, the cost does not exceed 0.05, whereas previously the cost was reaching values of 0.15.

We should note that in all experiments shown here the parameter c was chosen to be equal to zero. Experiments with different values of c , such as 0.01, showed the same results, and thus were omitted from this paper for conciseness.

Let us study how the noise levels affect the performance of the system with and without observer. Figure 7 shows both dependencies.

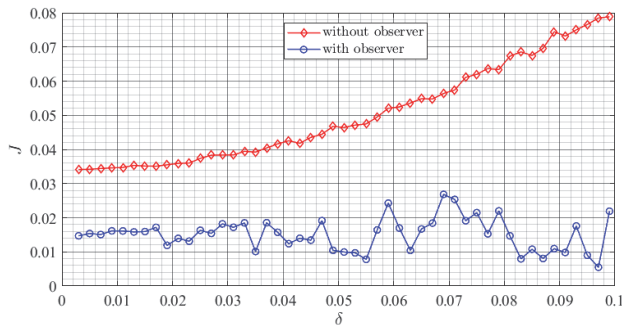


Fig. 7. Dependence of J on sensor noise δ (with the linear observer, $c=0, \sigma=0.05$).

Figure 7 demonstrates that the cost increases with the noise level when observer is not used, while introducing the observer makes the performance of the system seemingly independent from the noise.

It is well known, that high gain feedback control is more sensitive to sensor noise and measurement inaccuracies, compared with low gain feedback. In order to study this relation we fix the sensor parameters at $c=0, \delta=0.05$ and $\sigma=0.05$, and vary gain matrices $\mathbf{K}_p = k_p \mathbf{I}$ and $\mathbf{K}_d = k_d \mathbf{I}$, while changing scalar coefficients k_p and k_d . Figures 8 and 9 illustrate the dependence of J on the scalar coefficients k_p and k_d for cases with and without observer.

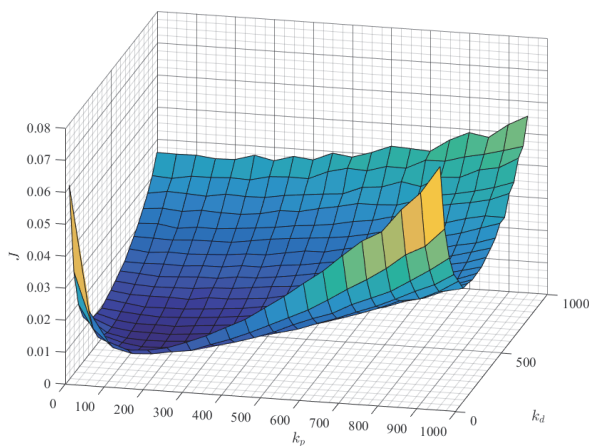


Fig. 8. Dependence of J on scalar coefficients k_p and k_d (without observer, $c=0, \delta=0.05, \sigma=0.05$).

Figure 8 shows that there exists a minimum value of J , obtained at $k_p=160$ and $k_d=150$. The optimal value

of J for that surface is $7.5 \cdot 10^{-3}$, while the mean value is 0.023. Increasing the values of the gain coefficients k_p and k_d worsens the performance of the system. This is an undesirable result, as higher controller gains allow higher motion precision.

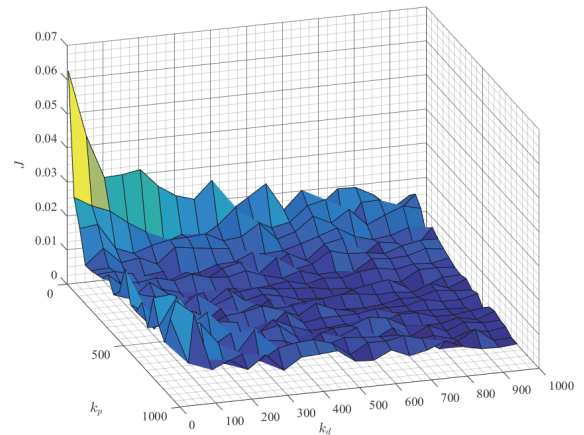


Fig. 9. Dependence of J on scalar coefficients k_p and k_d (with linear observer, $c=0, \delta=0.05, \sigma=0.05$).

Figure 9 shows that introducing the linear state observer allows to negate this problem, improving the performance of the system when the controller gains are high. The mean value of J for that surface is 0.011.

5 Simulations on a full scale robot

In this section, we present simulation results for the full robot model. This model takes into account mechanical constraints and the reaction forces that arise from them. In these experiments LQR-based controller discussed in [26] is used. Figure 10 shows the generalised velocities for the front legs of the robot for the case when $c=0, \delta=0.05$ and $\sigma=0.05$.

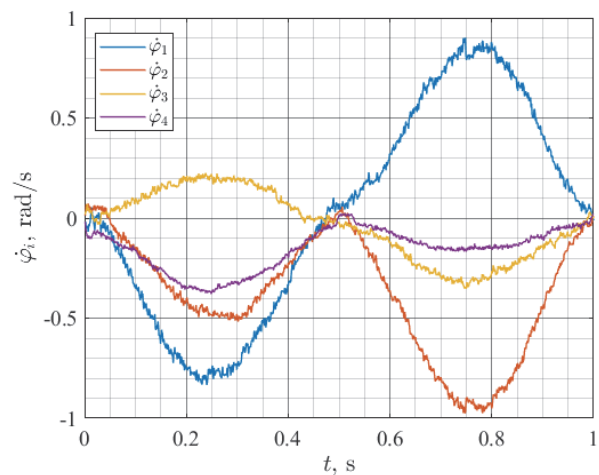


Fig. 10 Time functions for the generalized velocities for the full robot model (without observer, $c=0, \delta=0.05, \sigma=0.05$).

As we can see, the robot exhibits high frequency oscillations. This reflects the fact that the chosen LQR-based controller was tuned to produce high feedback

gains and thus the resulting control law is sensitive to the sensor noise. Figure 11 shows the same time functions obtained for the experiment with the linear state observer (7). The values of λ are quantized and sensor noise is added, same as it is done in (8). The corresponding constants c_λ , δ_λ and σ_λ have values $c_\lambda = 0$, $\delta_\lambda = 0.01$ and $\sigma_\lambda = 0.01$.

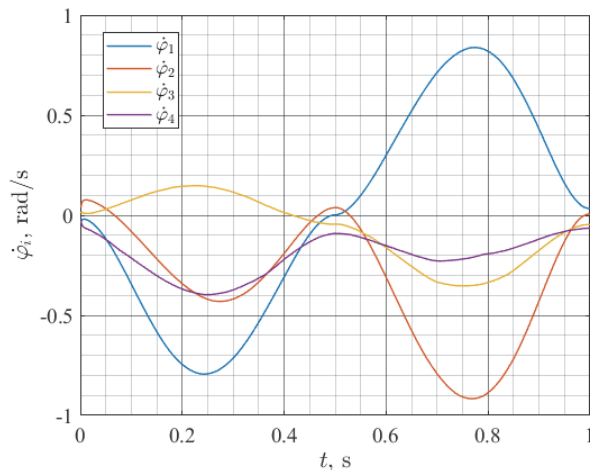


Fig. 11 Time functions for the generalized velocities for the full robot model (with the linear observer, $c = 0$, $\delta = 0.05$, $\sigma = 0.05$).

As we can see, the observer improved the behaviour of the robot, removing the tremor. Let us study how the behaviour of the observer depends on the quality of the reaction forces measurements. Figure 12 shows the dependence of J on the sensor resolution σ_λ .

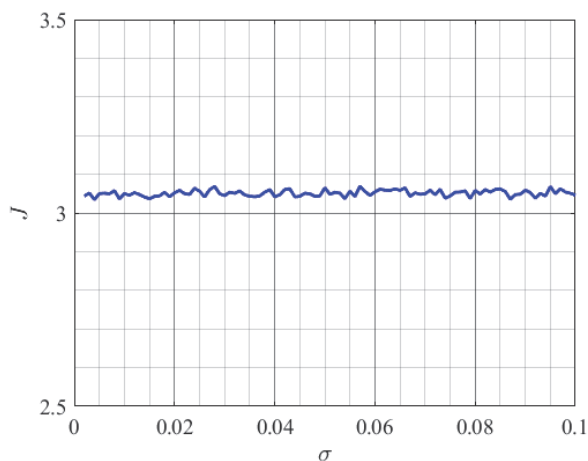


Fig. 12 Dependence of J on sensor resolution σ (without using the observer, $c_\lambda = 0$, $\delta_\lambda = 0.01$).

Figure 12 demonstrates that the performance function does now show dependence on the sensor resolution σ_λ in the chosen range. This is an important result, as it encourages the use of the state observer in systems with mechanical constraints, where the sensors measuring reaction forces have low accuracy.

We can also observe a significant increase in the values of the cost function. This is related to control errors, characteristic of the chosen controller. This is discussed in paper [14].

6 Conclusions

In this paper, the topic of state observation for in-pipe walking robots had been studied. These type of robots depend on accurate state information and accurate robot models, which should be derived from the erroneous sensor data, accessible to the robot. This signifies the question of choosing effective state observer. In this paper, an interactive linear state observer had been studied.

Simulations of motion of a single leg of the walking robot have showed that this type of observer significantly improves the behaviour of the control system in the presence of sensor noise and data quantization. It was demonstrated, that the introduction of the state observer allows the use of higher feedback controller gains for the case of a computed torque controller.

Simulations performed on the full robot model showed that the introduction of the state observer allows similar improvements. However, in order to use the state observer on the full robot mode, the information about reaction forces need to be included in the observer design. The observer demonstrated good performance even in the presence of noise and quantization of the reaction forces data.

The following research can be focused on designing an observer for reaction forces and on using the information of contact points locations in the state observer design.

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

References

1. N. Sh. Roslin et al, *Procedia Engineering* **41**, 1456–1462 (2012)
2. S.-g. Roh, and H. R. Choi, *IEEE transactions on robotics*, **21** (1), 1–17 (2005)
3. S.-g. Roh, J.-S. Lee, H. Moon, and H.R. Choi. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, 1724–1729 (2008)
4. S.M Ryew et al., *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, **3**, 1652–1657 (2000)
5. A.M. Bertetto, M. Ruggiu. *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, **2**, 1226–1231 (2001)
6. C. Anthierens, A. Ciftci, M. Betemps. *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'99)*, **2**, 968–972 (1999)
7. S. Jatsun, A. Yatsun, S. Savin, *Adaptive Mobile Robotics*, 713–719 (2012)
8. F. Pfeiffer, J. Eltze, H.-J. Weidemann, *Intelligent Automation & Soft Computing*, **1** (3), 307–323 (1995)

9. J.A. Gálvez, P.G. De Santos, F. Pfeiffer, IEEE/ASME Transactions on mechatronics, **6** (1) 26–35 (2001)
10. A. Zagler, F. Pfeiffer, Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA'03), **3**, 2954–2959 (2003)
11. R. Featherstone, *Rigid body dynamics algorithms* (Springer, 2014)
12. S. Savin, L. Vorochaeva, International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 1–6 (2017)
13. S. Savin, L. Vorochaeva, 2017 International Siberian Conference on Control and Communications (SIBCON), 1–6 (2017)
14. S. Savin, S. Jatsun, L. Vorochaeva, Dynamics of Systems, Mechanisms and Machines (Dynamics), 1–6 (2017)
15. S. Savin, S. Jatsun, L. Vorochaeva, MATEC Web of Conferences, **113**, 02016. (2017)
16. B. Yodwong et. al., 19th International Conference on Electrical Machines and Systems (ICEMS-2016), 1–6 (2016)
17. R. Cui, L. Chen, C. Yang, M. Chen. IEEE Transactions on Industrial Electronics, **64** (8), 6785–6795 (2017)
18. H. Yang, X. Fan, Y. Xia, C. Hua, Advanced Robotics, **30** (1), 68–78 (2016)
19. Y. Alothman, D. Gu, 2016 8th Computer Science and Electronic Engineering (CEECE), 168–173 (2016)
20. S. Jatsun, S. Savin, A. Yatsun, International Workshop on Medical and Service Robots, 305–317. (2016)
21. S. Jatsun, S. Savin, A. Yatsun, 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 239–244 (2016)
22. R. Ortega, and M. W. Spong, Automatica, **25** (6), 877–888 (1989)
23. S. Jatsun, S. Savin, B. Lushnikov, A. Yatsun, ITM Web of Conferences, **6**, 01001. (2016)
24. S. Jatsun, S. Savin, A. Yatsun, ROMANSY 21-Robot Design, Dynamics and Control, 361–368 (2016)
25. S. Jatsun, S. Savin, A. Yatsun, 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), 107–112 (2016)
26. S. Mason, N. Rotella, S. Schaal, L. Righetti. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), 63–68 (2016)