Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Geometric path following control of a rigid body based on the stabilization of sets ⋆

Yuri A. Kapitanyuk * **Sergey A. Chepinskiy** *
Aleksandr A. Kapitonov *

* *ITMO University*
49, Kronverkski prospekt, Saint-Petersburg, 197101, Russia
(e-mail: yura.kapitanyuk@gmail.com, Chepinsky_S@hotmail.com)

Abstract: The paper describes an approach to the development of the geometric path following control for a rigid body. Desired path of movement in the space is represented by an intersection of two implicit surfaces. Path following control problem is posed as a problem of maintaining the holonomic relationships between the system outputs. Control is synthesized using the differential geometrical method through nonlinear transformation of initial dynamic model. The main results presented are the model of spatial motion and relevant nonlinear control algorithms.

Keywords: Rigid body, path following, coordinates transformation, set stabilisation, nonlinear systems

1. INTRODUCTION

The paper considers the development of path following control system for a rigid body, that is, the problem of providing a motion along a given spatial path. With the advent of unmanned vehicles the path following control problem became even more urgent, because path following is an UAV major operating mode. Two approaches to the development of these systems are known [Aguiar et al. (2005), Nielsen et al. (2009)].

In the first one, a tracking system controlled by a reference model is developed [Breivik and Fossen (2005), Lee et al. (2010)]. The path is generally set by a time-dependent function, which leads to practical problems when the object motion is behind or ahead of the program due to parametric uncertainties or external disturbances. To solve this problem, the path should be parametrized by the length instead of time, and dynamics of this parameter should be introduced in the system model [Breivik and Fossen (2005)]. This method rather easily realizes the motion along polynomial curves, which provides better path planning and more accurate path following, however the resultant controller is rather lengthy.

An alternative approach is based on stabilization of invariant manifolds in state space based on feedback linearization [Nielsen et al. (2009),Hladio et al. (2013)] or passive-based control[El-Hawwary and Maggiore (2011),El-Hawwary and Maggiore (2013)]. Simply speaking, a transformation generating an attractor in state space is selected for the initial system. In path following context, the attractor is a desired path set in output coordinates. Then the designer should only stabilize this solution, which is much less demanding than creating a tracking system as in the first approach. As a control object, an autonomous robot is a multichannel nonlinear dynamic system. Control system of a mobile robot should generate control actions providing preset motion of the centre of mass in operating area. The one of the methods for synthesizing the control algorithms was proposed by I.V. Miroshnik. It is based on the second approach and implies nonlinear transformation of robot model to the task-oriented coordinate system, which makes it possible to reduce the complex multichannel control problem to several simple problems of compensation of linear and angular deviations and then to find adequate control laws using nonlinear stabilization [Fradkov et al. (1999), Kapitanyuk and Chepinsky (2013)].

Differentially geometric methods of nonlinear control theory [Fradkov et al. (1999), Kapitanyuk and Chepinsky (2013), Miroshnik and Nikiforov (1996), Miroshnik and Lyamin (1994), Miroshnik and Bobtsov (2000),Miroshnik and Sergeev (2001)] are used in the analysis method for these systems and synthesis of control algorithms solving the path following problem as a stabilization problem with respect to implicit space curve. This article deals with further development of task-oriented approach inspired by works[El-Hawwary and Maggiore (2011),Hua et al. (2013)]. The latest results based on feedback linearisation are presented in the first part of article. The further extension of task-oriented framework is described in the second part. Proposed design procedure base on the using of properties of passive systems. This article focuses directly on the synthesis of controllers without restricting the path planning method.

2. RIGID BODY MODEL AND STATEMENT OF CONTROL PROBLEM

Consider a fully actuated rigid body model illustrated in Fig. 1

$$\dot{x} = v, \tag{1}$$

⋆ This work was partially financially supported by Government of Russian Federation, Grant 074-U01
This work was supported by the Ministry of Education and Science of Russian Federation (Project 14.Z50.31.0031).

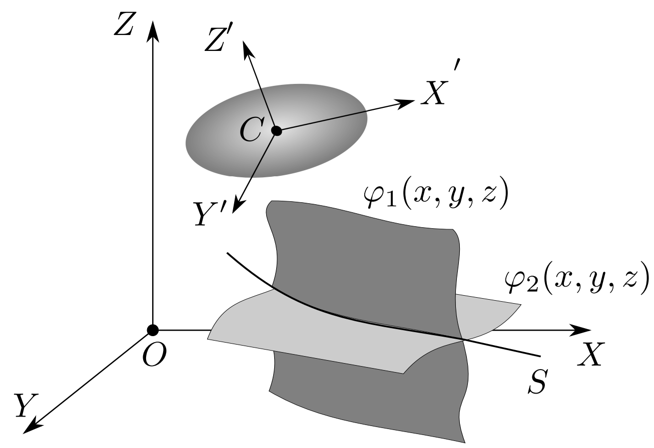

Fig. 1. Rigid body and desired trajectory in the space

$$m\dot{v} = F_c, \tag{2}$$

$$\dot{R}(\alpha) = S(\omega)R(\alpha), \tag{3}$$

$$J\dot{\omega} + \omega \times J\omega = M_c, \tag{4}$$

where $x = [x_{rb}, y_{rb}, z_{rb}]^T \in R^3$ is the Cartesian position vector of the center of mass C in the inertial reference frame XYZ, $v = [\dot{x}_{rb}, \dot{y}_{rb}, \dot{z}_{rb}]^T \in R^3$ is the velocity vector of the center of mass C in the inertial reference frame, $m \in R$ is a total mass of the rigid body, $F_c = [F_x, F_y, F_z]^T \in R^3$ is the vector of the control forces in the inertial reference frame, $\alpha = [\phi, \theta, \psi]^T \in R^3$ is the vector of Euler angels of the body-fixed frame in the inertial reference frame with yaw, pitch and roll angels respectively, $R(\alpha) \in SO(3)$ is the rotation matrix from the body-fixed frame to the inertial frame, $\omega = [\omega_\phi, \omega_\theta, \omega_\psi]^T \in R^3$ is the vector of angular velocities in the body-fixed frame, $M_c = [M_\phi, M_\theta, M_\psi]^T \in R^3$ is the vector of the control moments in the body-fixed frame, $S(\omega) \in SO(3)$ is the skew symmetric matrix with structure:

$$S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}. \tag{5}$$

The rotation matrix can be expressed through Euler angels as

$$R(\alpha) = R_3(\psi)R_2(\theta)R_1(\phi) \tag{6}$$

where

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix},$$

$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

$$R_3(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The desired path is a smooth segment of curve S (see Fig. 1) described as an intersection of two implicit surfaces

$$\varphi_1(x, y, z) = 0 \cap \varphi_2(x, y, z) = 0, \tag{7}$$

where φ_1, φ_2 and ψ are smooth functions.

Tangential velocity along the curve S is defined as

$$\dot{s} = \frac{\nabla\varphi_1 \times \nabla\varphi_2}{\|\nabla\varphi_1 \times \nabla\varphi_2\|} v, \tag{8}$$

where $\times$ is the vector product and $\|\cdot\|$ is the vector norm.

It should be noted the description of a curve as a smooth geometrical object is not the only one possible, and the selection of functions (7) is ambiguous. Selection of functions $\varphi_1(x, y, z)$ and $\varphi_2(x, y, z)$ mostly limited by regularity condition [Fradkov et al. (1999)] implying that Jacobian matrix

$$\Upsilon(x, y, z) = \begin{bmatrix} \dfrac{\nabla\varphi_1 \times \nabla\varphi_2}{\|\nabla\varphi_1 \times \nabla\varphi_2\|} \\ \dfrac{\nabla\varphi_1}{\|\nabla\varphi_1\|} \\ \dfrac{\nabla\varphi_2}{\|\nabla\varphi_2\|} \end{bmatrix} \tag{9}$$

is not degenerate for any vector $x = [x_{rb}, y_{rb}, z_{rb}]^T$ belonging to curve S, i.e.

$$det\Upsilon(x, y, z) \neq 0.$$

Path following control problem is posed as a problem of maintaining the holonomic relationships between the system outputs $[x_{rb}, y_{rb}, z_{rb}]^T$ set in (7). It is augmented by the description of desired longitudinal motion of the point of the centre of mass of the rigid body along the desired path S usually set using the reference velocity of longitudinal motion $V^* = \dot{s}^*$

Consider the errors of the path following [Fradkov et al. (1999), Kapitanyuk and Chepinsky (2013)]. Violation of condition (7) is characterized by deviations

$$e_1 = \varphi_1(x, y, z) \tag{10}$$

$$e_2 = \varphi_2(x, y, z) \tag{11}$$

zeroed at manifold S.

Therefore, the path following control problem for the rigid body consists in determination of inputs $F_c = [F_x, F_y, F_z]^T$ in closed loop, which provides:

(a) stabilization of robot motion with respect to the curve S, which implies asymptotic zeroing of spatial deviation vectors e_1 and e_2;

(b) asymptotic zeroing of velocity error

$$\Delta V = V^* - \dot{s}. \tag{12}$$

(c) stabilization of the required robot orientation with respect to the curve S

3. TRANSLATION MOTION CONTROL

To synthesize the path following controllers we transform the system model with account for (1)-(4) to the task-based (path-based) form [Fradkov et al. (1999), Kapitanyuk and Chepinsky (2013)] with outputs s, e_1 and e_2. To do it, we differentiate (8), (10) and (11) with respect to time:

$$\begin{bmatrix} \dot{s} \\ \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \Upsilon(x, y, z)v. \tag{13}$$

Once more differentiate (13) with account for (2):

$$\begin{bmatrix} \ddot{s} \\ \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} = \dot{\Upsilon}(x,y,z)v + \Upsilon(x,y,z)\frac{F_c}{m}. \quad (14)$$

Now consider the virtual (task-based) controls:

$$\dot{\Upsilon}(x,y,z)v + \Upsilon(x,y,z)\frac{F_c}{m} = \begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix}. \quad (15)$$

Substitute (15) to (14) and obtain:

$$\begin{bmatrix} \ddot{s} \\ \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} = \begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix}. \quad (16)$$

Now select the controllers:

$$u_s = K_s \Delta \dot{s}, \quad (17)$$

$$u_{e1} = -K_{1e1}\dot{e}_1 - K_{2e1}e_1, \quad (18)$$

$$u_{e2} = -K_{1e2}\dot{e}_2 - K_{2e2}e_2, \quad (19)$$

where K_s, K_{1e1}, K_{2e1}, K_{1e2}, K_{2e2} are positive coefficients which provides the desired dynamics of asymptotic zeroing of Δs, e_1 and e_2 deviations and solves the problems (a) and (b).

Now we determine actual control actions F_c and finally obtain:

$$F_c = m\Upsilon(x,y,z)^{-1}\left(\begin{bmatrix} u_s \\ u_{e1} \\ u_{e2} \end{bmatrix} - \dot{\Upsilon}(x,y,z)v\right). \quad (20)$$

In this section, we have described the general method for synthesizing the controller of translation motion.

4. ATTITUDE CONTROL

Now focus on the solution of problem (c). Introduce the vector of angular errors $\delta = [\delta_\phi, \delta_\theta, \delta_\psi]^T \in R^3$ and the angular deviations matrix

$$R(\delta) = R(\alpha)R^T(\alpha^*)R^T(\Delta), \quad (21)$$

where $R(\alpha^*) \in SO(3)$ is the matrix of angular orientation of the body-fixed frame along the curve S in the point $[x_{rb}, y_{rb}, z_{rb}]^T$, $R^T(\Delta) \in SO(3)$ is the matrix of the desired angular orientation respect to the curve S.

For the stabilisation of the desired attitude and eliminating of the vector δ(or fulfilment of the identity $R(\delta) = I$) use approach described in work [Lee et al. (2010)]. Define the angular error function as

$$e_r = \frac{1}{2}\left(R(\delta) - R(\delta)^T\right)^\vee, \quad (22)$$

where *vee* map $\vee$ is the transformation $SO(3) \to R^3$.

Now define the angular speed error e_ω. Differentiating (21) with respect to time and the expression (3) and obtain the equation

$$\frac{d}{dt}R(\delta) = S(\dot{\delta})R(\delta) = e_\omega R(\delta), \quad (23)$$

$$\frac{d}{dt}R(\delta) = S(\omega)R(\delta) - R(\alpha)R^T(\alpha^*)S(\omega^*)R^T(\Delta), \quad (24)$$

Use the property of skew symmetric matrix $RS(\omega)R^T = S(R\omega)$ obtain the final expression

$$\frac{d}{dt}R(\delta) = \left(S(\omega) - S(R(\alpha)R^T(\alpha^*)\omega^*)\right)R(\delta), \quad (25)$$

and

$$e_\omega = \omega - R(\alpha)R^T(\alpha^*)\omega^*. \quad (26)$$

Differentiating (27) with account for (3)

$$\dot{e}_\omega = \frac{1}{J}\left(M - \omega \times J\omega\right) + a_d \quad (27)$$

where

$$a_d = -S(\omega)R(\alpha)R^T(\alpha^*)\omega^* + R(\alpha)R^T(\alpha^*)\dot{\omega}^* \quad (28)$$

Resulting attitude controller has form

$$M_c = \omega \times J\omega - Ja_d - K_R e_r - K_\omega e_\omega. \quad (29)$$

where K_R and K_ω are positive constants.

In a common case for the matrix $R(\alpha^*)$ can use the orthogonal matrix based on Jacobian (9)

$$R(\alpha^*) = T(x,y,z) = \begin{bmatrix} \dfrac{\nabla\varphi_1 \times \nabla\varphi_2}{\|\nabla\varphi_1 \times \nabla\varphi_2\|} \\ \dfrac{\nabla\varphi_1}{\|\nabla\varphi_1\|} \\ \dfrac{(\nabla\varphi_1 \times \nabla\varphi_2) \times \nabla\varphi_1}{\|(\nabla\varphi_1 \times \nabla\varphi_2) \times \nabla\varphi_1\|} \end{bmatrix}. \quad (30)$$

The vector of desire angular speed can found from the Frenet-like equation [Fradkov et al. (1999)]:

$$\dot{T}(x,y,z) = \dot{s}S(\xi)T(x,y,z),$$

$$S(\xi) = \begin{bmatrix} 0 & \tau & 0 \\ -\tau & 0 & \kappa \\ 0 & -\kappa & 0 \end{bmatrix},$$

where $\xi = [\tau, 0, \kappa]^T \in R^3$, κ - curvature of curve S in the point(x,y,z), τ - torsion of curve S in the point (x, y, z).

Then equations (27), (28) and (30) transform to

$$e_\omega = \omega - \dot{s}R(\alpha)A^T(x,y,z)\xi,$$

$$\begin{aligned} a_d = &-\ddot{s}R(\alpha)T^T(x,y,z)\xi - \\ &-\dot{s}S(\omega)R(\alpha)T^T(x,y,z)\xi + \\ &+\dot{s}^2R(\alpha)A^T(x,y,z)S(\xi)\xi - \\ &-\dot{s}R(\alpha)A^T(x,y,z)\dot{\xi}. \end{aligned}$$

In this section, we have described the general method for synthesizing the controller of attitude motion which solves the problems (c). Now focus on particular cases popular in practice.

5. PASSIVE BASED PATH FOLLOWING

Consider the same model (1)-(4) and storage function V

$$V = \frac{1}{2}|\dot{x} - v_d|^2 + \frac{1}{2}|\omega - \omega_d|^2 \quad (31)$$

Differentiating (31) with respect to time and obtain the equation

$$\dot{V} = (\dot{x} - v_d)^T(\ddot{x} - \dot{v}_d) + (\omega - \omega_d)^T(\dot{\omega} - \dot{\omega}_d) =$$
$$= (\dot{x} - v_d)^T(\frac{1}{m}F - \dot{v}_d) +$$
$$+(\omega - \omega_d)^T(-J^{-1}\omega \times J\omega + J^{-1}M_c - \dot{\omega}_d)$$

Now choose the control inputs F and M_c in the forms

$$F = u_{pF} + u_F,$$
$$M_c = u_{pM} + u_M,$$

where $u_{pF} = \dot{v}_d$ and $u_{pM} = \omega \times J\omega + J\dot{\omega}_d$ are control inputs, which make system passive. In the result we obtain

$$\dot{V} = (\dot{x} - v_d)^T u_F + (\omega - \omega_d)^T u_M = y^T u.$$

It means that system (1)-(4) is passive from inputs

$$u = [u_F^T, u_M^T]^T$$

to outputs

$$y = [(\dot{x} - v_d)^T, (\omega - \omega_d)^T]^T.$$

For the stabilisation of passive system we can use ordinary feedback control law

$$u = -Ky$$

where K is diagonal matrix of scalar coefficients $k_1, .., k_6$ Now we can use variables v_d and ω_d as new "virtual" inputs and stabilise the desired manifolds for the translation motion

$$\varphi_1(x, y, z) = 0 \cap \varphi_2(x, y, z) = 0,$$

and desire orientation presented by relations [Hua et al. (2013)]

$$(1 - n^T n_d),$$

where n is a vector of current orientation and n_d is a vector of desired orientation Consider Lyapunov function

$$W = \frac{1}{2}\varphi_1^2 + \frac{1}{2}\varphi_2^2 + \frac{1}{2}(1 - n^T n_c)^2 \tag{32}$$

Take the derivative of the function (32) with respect to time

$$\dot{W} = \varphi_1 \left(\frac{\partial \varphi_1}{\partial x}\right)^T \dot{x} + \varphi_2 \left(\frac{\partial \varphi_2}{\partial x}\right)^T \dot{x} -$$
$$-\frac{d}{dt}n^T n_c - n^T \frac{d}{dt} n_d =$$
$$= \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T v_d +$$
$$+(1 - n^T n_c)(n^T S(\omega_d) n_c - n^T S(\omega_c) n_c) =$$
$$= \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T v_d +$$
$$+(1 - n^T n_c) n^T S(\omega_d - \omega_c) n_c =$$
$$= \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T v_d +$$
$$+(1 - n^T n_c) n^T S^T(n_c)(\omega_d - \omega_c)$$

Then for $v_d = u_\tau + u_\varphi$ and $\omega_d = \omega_c + u_\omega$ obtain

$$\dot{W} = \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T \Upsilon^{-1}[V^*, 0, 0]^T +$$
$$+ \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T u_\varphi +$$
$$+(1 - n^T n_c) n^T S^T(n_c) u_\omega$$

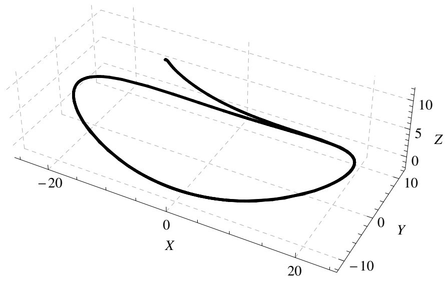

Fig. 2. Motion along space curve

Select the control inputs u_φ and u_ω as

$$u_\varphi = -K_\varphi \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)$$
$$u_\omega = -S(n_c) n k_\omega (1 - n^T n_c)$$

Then equation of derivative of Lyapunov function becomes

$$\dot{W} = -\left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right)^T K_\varphi \left(\varphi_1 \frac{\partial \varphi_1}{\partial x} + \varphi_2 \frac{\partial \varphi_2}{\partial x}\right) -$$
$$-k_\omega (1 - n^T n_c)^2 < 0,$$

where K_φ is the positive defined matrix of constant coefficients and k_ω is a strictly positive constant. This result means that our system is asymptotic stable with domain of attraction $(-\pi, \pi)$.

6. NUMERICAL EXAMPLE

For the illustration of the rigid body motion we have considered two cases.

At first, we was modelling the common control law based on feedback linearisation approach for the rigid body with $m = 1$, $J = 1$. The desire path S is an intersection of two implicit surfaces such as elliptic cylinder and parabola

$$(\varphi_1(x, y, z) = 0.2x^2 + y^2 - R^2) \cap (\varphi_2(x, y, z) = z + 0.05y^2 - 5)$$

Initial position of the centre C of the rigid body is $x_0 = [-10, 5, 10]^T$ and initial orientation is $\alpha_0 = [3, 2, 1]^T$

Parameters of the controller is presented below

$$K_{e11} = 1, K_{e12} = 10;$$
$$K_{e21} = 1, K_{e22} = 10;$$
$$K_R = 20, K_\omega = 50.$$

Desire speed along the trajectory $\dot{s}^* = 1$

For the attitude control was used simplified version of control law (30) without feed-forward term a_d

$$M_c = \omega \times J\omega - K_R e_r - K_\omega e_\omega.$$

The results of modelling is presented on figures (2)-(8).

In the second case, was modelling the passivity based proposed control law for the rigid body spatial motion for the same desire path S. Desire speed along the trajectory $\dot{s}^* = 1$. Initial orientation describes by vector $n = [1, 0, 0]$. The desired orientation set by vector $n_c = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]$. Parameters of controllers is represented below

$$K = diag\,[\,0.1\ 0.1\ 0.1\ 10\ 10\ 10\,]$$
$$K_\varphi = diag\,[\,2\ 2\ 2\,]$$
$$k_\omega = 8$$

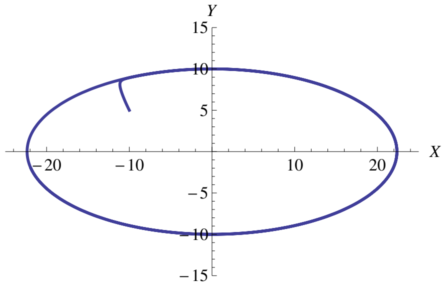

Fig. 3. Projection on XY plane

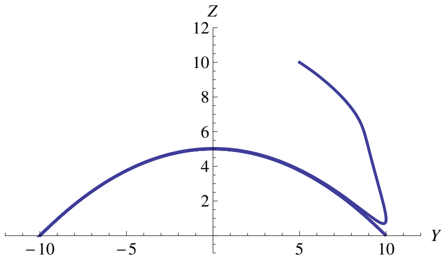

Fig. 4. Projection on YZ plane

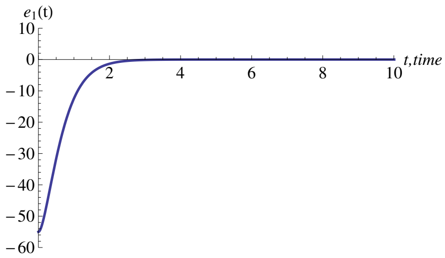

Fig. 5. Position error $e_1 = \varphi_1(x, y, z)$

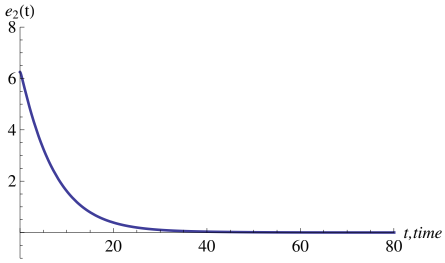

Fig. 6. Position error $e_2 = \varphi_2(x, y, z)$

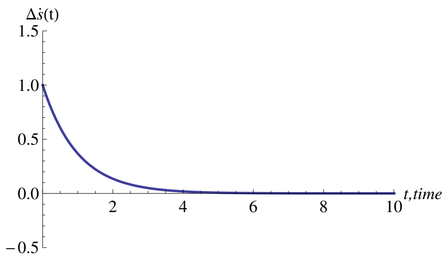

Fig. 7. Speed error $\Delta V = \dot{s}^* - \dot{s}$

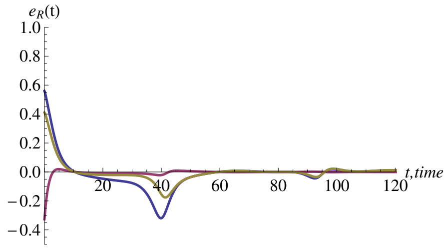

Fig. 8. Angular error e_r

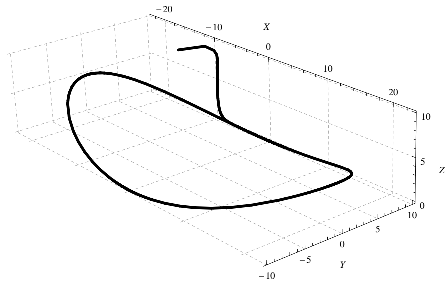

Fig. 9. Motion along space curve

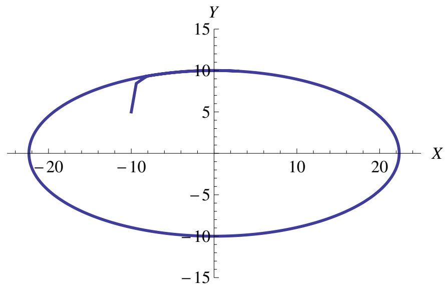

Fig. 10. Projection on XY plane

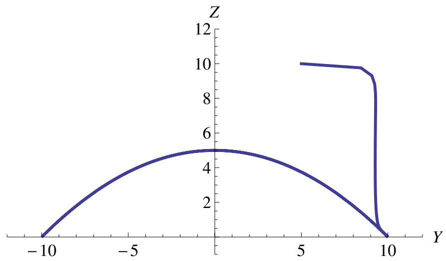

Fig. 11. Projection on YZ plane

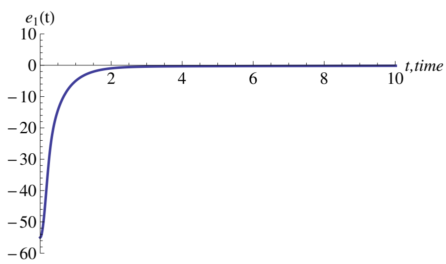

Fig. 12. Position error $e_1 = \varphi_1(x, y, z)$

The result of modelling is presented on figures (9)-(16) We can't direct compare this two approach yet, for the detailed analysis we should deeper research this two method for identify of possible advantages and disadvantages. This is the main task for further work.

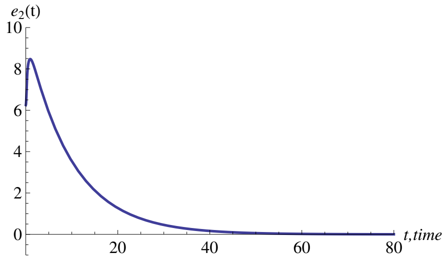

Fig. 13. Position error $e_2 = \varphi_2(x, y, z)$

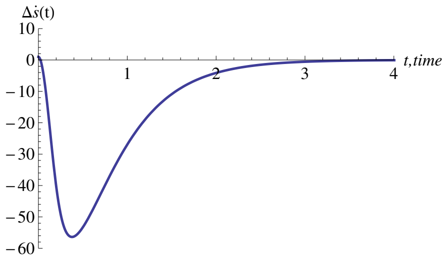

Fig. 14. Speed error $\Delta V = \dot{s}^* - \dot{s}$

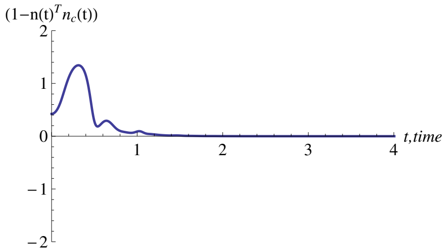

Fig. 15. Angular error $(1 - n^T(t)n(t)_c)$

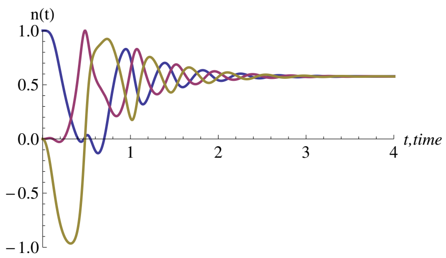

Fig. 16. Orientation of the reference axis $n(t)$

7. CONCLUSIONS

The proposed control algorithms can be helpful in development of path following control systems for mobile robots (underwater or airborne robots). This method is rather computational complex in common form, but with additional assumptions can find the simple controllers. The primary of the future research is the detailed compression of the represented approach. Also it would be interesting to apply this method to adequate robot dynamic models, for example quadrotor UAV. Moreover the performance of proposed controllers in the presence of parametric uncertainties and external disturbances, and development version of this controller without velocity measuring should be the subject to further analysis.

REFERENCES

Aguiar, A., Hespanha, J., and Kokotovic, P. (2005). Path-following for nonminimum phase systems removes performance limitations. *Automatic Control, IEEE Transactions on*, 50(2), 234–239. doi: 10.1109/TAC.2004.841924.

Breivik, M. and Fossen, T. (2005). Principles of guidance-based path following in 2d and 3d. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, 627–634. doi: 10.1109/CDC.2005.1582226.

El-Hawwary, M.I. and Maggiore, M. (2011). Case studies on passivity-based stabilisation of closed sets. *International Journal of Control*, 84(2), 336–350.

El-Hawwary, M.I. and Maggiore, M. (2013). Reduction theorems for stability of closed sets with application to backstepping control design. *Automatica*, 49(1), 214–222.

Fradkov, A., Miroshnik, I., and Nikiforov, V. (1999). *Nonlinear and Adaptive Control of Complex Systems*. Mathematics and Its Applications. Springer.

Hladio, A., Nielsen, C., and Wang, D. (2013). Path following for a class of mechanical systems. *Control Systems Technology, IEEE Transactions on*, 21(6), 2380–2390.

Hua, M.D., Hamel, T., Morin, P., and Samson, C. (2013). Introduction to feedback control of underactuated vtol vehicles: A review of basic control design ideas and principles. *IEEE Control Systems Magazine*, 33(1), 61–75. doi:10.1109/MCS.2012.2225931.

Kapitanyuk, Y. and Chepinsky, S. (2013). Control of mobile robot following a piecewise-smooth path. *Gyroscopy and Navigation*, 4(4), 198–203. doi:10.1134/S207510871304007X. URL `http://dx.doi.org/10.1134/S207510871304007X`.

Lee, T., Leoky, M., and McClamroch, N. (2010). Geometric tracking control of a quadrotor uav on se(3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, 5420–5425. doi:10.1109/CDC.2010.5717652.

Miroshnik, I. and Bobtsov, A. (2000). Stabilization of motions of multi-pendulum systems. In *Control of Oscillations and Chaos, 2000. Proceedings. 2000 2nd International Conference*, volume 1, 22–25 vol.1. doi: 10.1109/COC.2000.873499.

Miroshnik, I. and Lyamin, A. (1994). Nonlinear control of multidrive vehicular robots. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, 79–80 vol.1. doi:10.1109/CCA.1994.381246.

Miroshnik, I. and Nikiforov, V. (1996). Trajectory motion control and coordination of multilink robots. *Prepr. 13th IFAC World Congress*, A, 361–366.

Miroshnik, I. and Sergeev, K. (2001). Nonlinear control of robot spatial motion in the dynamic environments. *Proc. Int. IEEE Conf. On Advanced Intel. Mechatronics*, 2, 1303–1306.

Nielsen, C., Fulford, C., and Maggiore, M. (2009). Path following using transverse feedback linearization: Application to a maglev positioning system. In *American Control Conference, 2009. ACC '09.*, 3045–3050. doi: 10.1109/ACC.2009.5159998.