SIMULATIONAL STUDY OF A HYDRAULICLY DRIVEN PARALLEL MANIPULATOR CONTROL SYSTEM

Janusz Frączek, Marek Wojtyra

Warsaw University of Technology
Institute of Aeronautics and Applied Mechanics
Nowowiejska Street 24, 00-665 Warsaw, Poland
e-mail: jfraczek@meil.pw.edu.pl, mwojtyra@meil.pw.edu.pl

Ioannis Davliakos, Evangelos Papadopoulos

National Technical University of Athens
Department of Mechanical Engineering
15780 Athens, Greece
e-mail: gdavliak@central.ntua.gr, egpapado@central.ntua.gr

Abstract

A simulational model of 6 dof Stewart platform type parallel manipulator is presented. The manipulator is driven by electrohydraulic actuators. The developed control scheme employs the dynamic and hydraulic model of the system. The manipulator model consists of the rigid body equations of motion and the hydraulic dynamics of the main elements. Friction is included in the model. Two well-known software packages, one designed to perform control system simulation and the other dedicated to multibody simulation, are simultaneously used to conduct the study. The aim of the study is to check if it is the impact of dynamics model accuracy on the quality of control process. The simplifications of the model and the problems with finding accurate values of its parameters (especially friction parameters) are considered. The study helps to predict the possible results of inaccurate determination of the crucial model parameters. The simulation results show, that inverse dynamics model simplifications consisting in neglecting the mass of actuators have relatively little influence on the control quality. The effects of actuators masses neglecting can be reduced by appropriate changes in the modelled mass of the platform. The study shows that friction effects should be introduced to the inverse dynamics model and the friction parameters should be identified with possibly big accuracy. The parameters describing stiction-friction transition effects are the crucial ones.

Keywords: robotics, control, dynamics, simulation, parallel manipulators, inverse dynamics models

1. Introduction

Parallel manipulators are quite often driven by electrohydraulic actuators. Hydraulics drives are commonly used due to their ability to produce large forces at high speeds, their high durability and stiffness, and their rapid response. Hydraulic systems differ from electromechanical ones, in that the force or torque output is not proportional to actuator current. As a result, controllers that have been designed for electrically driven robot control cannot be used. A brief overview of control techniques used for electrohydraulic servosystems can be found in [3].

In contemporary parallel robots the position control – widely used in earlier designs – is replaced by a model-based control. Due to high sampling frequency of control systems, the driving forces calculations must be performed quickly. Therefore the inverse dynamics model of manipulator is usually simplified, and thus some aspects of manipulator motion are not represented in their full complexity.

Hydraulic actuator output force is a highly nonlinear function of control current, which causes additional problems. A numerical model of hydraulic actuator and servovalve dynamics must be incorporated into the control system. Accurate values of such a model parameters (e.g. actuator friction parameters) are difficult to find.

The aim of presented study was to check if it was the impact of dynamics model accuracy on the quality of control process. The simplifications of the model and the problems with finding accurate values of its parameters (especially friction parameters) were considered. The results of this study should help to decide how big could be the simplifications of the dynamics model. The study should also help predict if they are the possible results of inaccurate determination of the crucial model parameters. The investigated control scheme employs the dynamic and hydraulic model of the system. The manipulator model consists of the rigid body equations of motion and the hydraulic dynamics of the main elements. Friction is included in the model.

Two well-known software packages, one designed to perform control system simulation and the other dedicated to multibody simulation, have been used to conduct the study. These packages were collaborating during simulations – both the programs performed all calculations simultaneously.
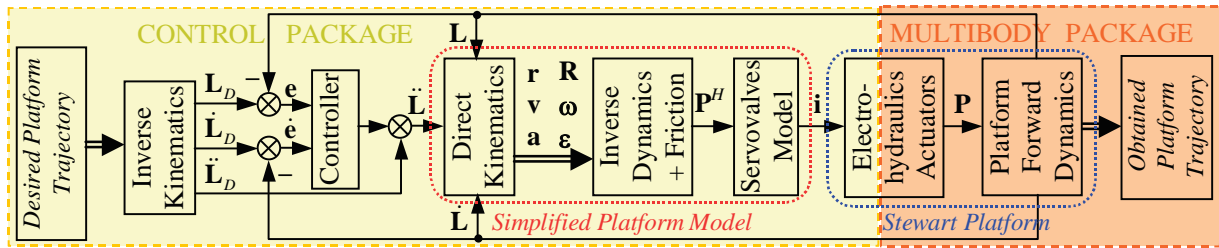

Fig. 1. Schematic view of the simulational model

The manipulator forward dynamics has been modelled using a multibody package, which automatically generates and solves the equations of motion. Therefore it is relatively easy to introduce changes into the model and to take into account various factors, for example joint friction or interactions with environment. There is no need for tedious and difficult process of deriving and programming manually the necessary equations. Therefore, crude simplifications of the multibody model are not required. The additional benefit of using multibody package was the possibility to create and watch animations of the manipulator in motion.

The control system and the electrohydraulic servovalves have been modelled in the control software package. Since the model-based control scheme is adopted, the inverse dynamics problem must be solved within the control system. Friction effects are included in the inverse dynamics model. The model utilized by the control system is simplified, to enable fast calculations.

The model scheme is shown in Fig. 1. It is worth noting, that hydraulic actuator dynamics is included in the control package, whereas the rigid body mechanics is modelled in the multibody package.

2. Manipulator kinematics

The kinematic diagram of manipulator is presented in Fig. 2. To simplify the picture, only one hydraulic actuator is presented.

The coordinates of position vectors $\mathbf{d}_j$ ($j = 1,\ldots, 6$) are constant in the π_0 frame, and coordinates of position vectors $\mathbf{s}_j^{(1)}$ ($j = 1,\ldots, 6$) are constant in the π_1 frame.

2.1. Desired trajectory generation

The position of local frame π_1 (established on the moving platform) in the global frame π_0 (established on the basis of manipulator) is described by vector $\mathbf{r}$, and the orientation of π_1 frame with respect to π_0 frame is given by three Euler (z–x'–z'') angles: φ_1, φ_2, φ_3. The coordinates of vector $\mathbf{r}$ and angles φ_1, φ_2, φ_3 are assumed functions of time.

For given values of Euler angles, the direction cosine matrix (rotation matrix) describing the π_1 frame orientation with respect to the π_0 frame, can be calculated as follows:
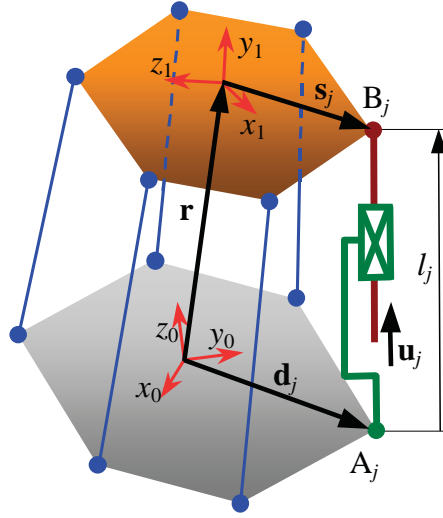

Fig. 2. Simplified kinematic scheme of the manipulator

$$\mathbf{R} = \mathbf{R}_z(\varphi_1)\,\mathbf{R}_x(\varphi_2)\,\mathbf{R}_z(\varphi_3) = \begin{bmatrix} \cos\varphi_1 & -\sin\varphi_1 & 0 \\ \sin\varphi_1 & \cos\varphi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_2 & -\sin\varphi_2 \\ 0 & \sin\varphi_2 & \cos\varphi_2 \end{bmatrix}\begin{bmatrix} \cos\varphi_3 & -\sin\varphi_3 & 0 \\ \sin\varphi_3 & \cos\varphi_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\varphi_3\cos\varphi_1 - \sin\varphi_3\cos\varphi_2\sin\varphi_1 & -\sin\varphi_3\cos\varphi_1 - \cos\varphi_3\cos\varphi_2\sin\varphi_1 & \sin\varphi_2\sin\varphi_1 \\ \sin\varphi_3\cos\varphi_2\cos\varphi_1 + \cos\varphi_3\sin\varphi_1 & \cos\varphi_3\cos\varphi_2\cos\varphi_1 - \sin\varphi_3\sin\varphi_1 & -\sin\varphi_2\cos\varphi_1 \\ \sin\varphi_3\sin\varphi_2 & \cos\varphi_3\sin\varphi_2 & \cos\varphi_2 \end{bmatrix}. \tag{1}$$

The linear velocity of π_1 origin with respect to π_0 is obtained by differentiation of $\mathbf{r}$ with respect to time:

$$\mathbf{v} = \dot{\mathbf{r}}, \tag{2}$$

and the angular velocity of π_1 frame can be calculated as follows:

$$\boldsymbol{\omega} = \mathbf{E}(\varphi_1,\varphi_2,\varphi_3)\begin{bmatrix}\dot\varphi_1 \\ \dot\varphi_2 \\ \dot\varphi_3\end{bmatrix} = \mathbf{E}(\boldsymbol{\varphi})\,\dot{\boldsymbol{\varphi}}, \qquad \boldsymbol{\varphi} = \begin{bmatrix}\alpha \\ \beta \\ \gamma\end{bmatrix}, \qquad \mathbf{E}(\boldsymbol{\varphi}) = \begin{bmatrix} 0 & \cos\varphi_1 & \sin\varphi_1\sin\varphi_2 \\ 0 & \sin\varphi_1 & -\cos\varphi_1\sin\varphi_2 \\ 1 & 0 & \cos\varphi_2 \end{bmatrix}. \tag{3}$$

The linear acceleration of π_1 frame origin is obtained by differentiation of velocity vector:

$$\mathbf{a} = \dot{\mathbf{v}} = \ddot{\mathbf{r}}, \tag{4}$$

and the angular velocity is given by:

$$\boldsymbol{\varepsilon} = \dot{\boldsymbol{\omega}} = \dot{\mathbf{E}}\dot{\boldsymbol{\varphi}} + \mathbf{E}\ddot{\boldsymbol{\varphi}}, \qquad \dot{\mathbf{E}}(\boldsymbol{\varphi},\dot{\boldsymbol{\varphi}}) = \begin{bmatrix} 0 & -\dot\varphi_1\sin\varphi_1 & \dot\varphi_1\cos\varphi_1\sin\varphi_2 + \dot\varphi_2\sin\varphi_1\cos\varphi_2 \\ 0 & \dot\varphi_1\cos\varphi_1 & \dot\varphi_1\sin\varphi_1\sin\varphi_2 - \dot\varphi_2\cos\varphi_1\cos\varphi_2 \\ 0 & 0 & -\dot\varphi_2\sin\varphi_2 \end{bmatrix}. \tag{5}$$

2.2. Inverse kinematics

The inverse problem of kinematics consists in searching for the actuators motion (lengths, velocities and accelerations), when the platform motion is given (position, velocity, acceleration).

Since in the inverse kinematics problem vector $\mathbf{r}$ and matrix $\mathbf{R}$ are given, vector $\mathbf{l}_j$ from point A_j to point B_j can be calculated as:

$$\mathbf{l}_j = \mathbf{r} + \mathbf{R}\mathbf{s}_j^{(1)} - \mathbf{d}_j. \tag{6}$$

The length of $\mathbf{l}_i$ can be obtained as:

$$l_j = \sqrt{\mathbf{l}_j^T \mathbf{l}_j}. \tag{7}$$

Let a unit vector of actuator j be defined as:

$$\mathbf{u}_j = \mathbf{l}_j / l_j. \tag{8}$$

Let the lengths of all six actuators form an algebraic vector $\mathbf{L}$:

$$\mathbf{L} = \begin{bmatrix} l_1 & \cdots & l_6 \end{bmatrix}^T. \tag{9}$$

Differentiating Eq. (6) with respect to time and taking into account the rotation matrix properties, results in:

$$\dot{\mathbf{l}}_j = \dot{\mathbf{r}} + \dot{\mathbf{R}}\mathbf{s}_j^{(1)} = \mathbf{v} + \tilde{\boldsymbol{\omega}}\mathbf{R}\mathbf{s}_j^{(1)} = \mathbf{v} + \tilde{\boldsymbol{\omega}}\mathbf{s}_j. \tag{10}$$

In the above equation $\mathbf{s}_j$ is the vector pointing form frame π_1 origin to point B_j, its coordinates are resolved in frame π_0 ($\mathbf{s}_j = \mathbf{R}\mathbf{s}_j^{(1)}$).

The length of unit vector $\mathbf{u}_j$ is equal to one. Thus, the time derivative of the unit vector $\mathbf{u}_j$ is orthogonal to the vector itself. It can be denoted as follows:

$$\mathbf{u}_j^T \mathbf{u}_j = 1, \tag{11}$$

$$\mathbf{u}_j^T \dot{\mathbf{u}}_j = 0. \tag{12}$$

The $\mathbf{l}_j$ vector can be expressed as:

$$\mathbf{l}_j = l_j \mathbf{u}_j. \tag{13}$$

Differentiating Eq. (13) with respect to time we obtain:

$$\dot{\mathbf{l}}_j = \dot{l}_j \mathbf{u}_j + l_j \dot{\mathbf{u}}_j. \tag{14}$$

Premultiplying the above equation by $\mathbf{u}_j^T$ and utilizing Eq. (12) and Eq. (11) yields:

$$\mathbf{u}_j^T \dot{\mathbf{l}}_j = \dot{l}_j \mathbf{u}_j^T \mathbf{u}_j + l_j \mathbf{u}_j^T \dot{\mathbf{u}}_j = \dot{l}_j. \tag{15}$$

Substituting Eq. (10) into Eq. (15) results in:

$$\dot{l}_j = \mathbf{u}_j^T \dot{\mathbf{l}}_j = \mathbf{u}_j^T \left(\mathbf{v} + \tilde{\boldsymbol{\omega}}\mathbf{s}_j \right) = \mathbf{u}_j^T \mathbf{v} - \mathbf{u}_j^T \tilde{\mathbf{s}}_j \boldsymbol{\omega} = \mathbf{J}_j \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}. \tag{16}$$

In the above equation $\mathbf{J}_j$ denotes j-th row of manipulator Jacobian matrix:

$$\mathbf{J}_j = \begin{bmatrix} \mathbf{u}_j^T & -\mathbf{u}_j^T \tilde{\mathbf{s}}_j \end{bmatrix}. \tag{17}$$

Unknown actuator velocities can be calculated using Eq (16). Let the velocities of all six actuators form an algebraic vector $\dot{\mathbf{L}}$:

$$\dot{\mathbf{L}} = \begin{bmatrix} \dot{l}_1 & \cdots & \dot{l}_6 \end{bmatrix}^T. \tag{18}$$

Differentiating Eq. (10) with respect to time yields:

$$\ddot{\mathbf{l}}_j = \dot{\mathbf{v}} + \dot{\tilde{\boldsymbol{\omega}}}\mathbf{R}\mathbf{s}_j^{(1)} + \tilde{\boldsymbol{\omega}}\dot{\mathbf{R}}\mathbf{s}_j^{(1)} = \dot{\mathbf{v}} + \dot{\tilde{\boldsymbol{\omega}}}\mathbf{R}\mathbf{s}_j^{(1)} + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{R}\mathbf{s}_j^{(1)} = \mathbf{a} + \tilde{\boldsymbol{\varepsilon}}\mathbf{s}_j + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}_j. \tag{19}$$

Differentiating (15) with respect to time results in:

$$\ddot{l}_j = \dot{\mathbf{u}}_j^T \dot{\mathbf{l}}_j + \mathbf{u}_j^T \ddot{\mathbf{l}}_j . \tag{20}$$

Vector $\dot{\mathbf{u}}_j$ can be calculated using Eq. (14):

$$\dot{\mathbf{u}}_j = \frac{1}{l_j}\left(\dot{\mathbf{l}}_j - \dot{l}_j \mathbf{u}_j\right). \tag{21}$$

Substitution of Eq. (19) into Eq. (20) gives:

$$\ddot{l}_j = \dot{\mathbf{u}}_j^T \dot{\mathbf{l}}_j + \mathbf{u}_j^T\left(\mathbf{a} + \tilde{\boldsymbol{\varepsilon}}\mathbf{s}_j + \tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}_j\right) = \mathbf{u}_j^T\mathbf{a} - \mathbf{u}_j^T\tilde{\mathbf{s}}_j\boldsymbol{\varepsilon} + \dot{\mathbf{u}}_j^T\dot{\mathbf{l}}_j + \mathbf{u}_j^T\tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}_j = \mathbf{J}_j\begin{bmatrix}\mathbf{a}\\\boldsymbol{\varepsilon}\end{bmatrix} + \dot{\mathbf{u}}_j^T\dot{\mathbf{l}}_j + \mathbf{u}_i^T\tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{s}_j . \tag{22}$$

Unknown actuator accelerations can be calculated using the above equation. Let the accelerations of all six actuators form an algebraic vector $\ddot{\mathbf{L}}$:

$$\ddot{\mathbf{L}} = \begin{bmatrix}\ddot{l}_1 & \cdots & \ddot{l}_6\end{bmatrix}^T . \tag{23}$$

2.3. Direct kinematics

The direct problem of kinematics consists in searching for the platform motion (position, velocity, acceleration), when the actuators motion is given (lengths, velocities and accelerations).

Numerical method will be employed to solve the position problem. To simplify the notation it is convenient to have homogenous names of unknown parameters describing platform position and orientation (vector $\mathbf{r}$ coordinates and Euler angles corresponding to matrix $\mathbf{R}$). Thus, let us introduce the following symbols:

$$\mathbf{q} \equiv \begin{bmatrix}q_1 & q_2 & q_3 & q_4 & q_5 & q_6\end{bmatrix}^T \equiv \begin{bmatrix}r_x & r_y & r_z & \varphi_1 & \varphi_2 & \varphi_3\end{bmatrix}^T \equiv \begin{bmatrix}\mathbf{r}^T & \boldsymbol{\varphi}^T\end{bmatrix}^T . \tag{24}$$

Substituting Eq. (6) into square of Eq. (7) yields:

$$l_j^2 = \mathbf{l}_j^T\mathbf{l}_j = \left(\mathbf{r} + \mathbf{R}\mathbf{s}_j^{(1)} - \mathbf{d}_j\right)^T\left(\mathbf{r} + \mathbf{R}\mathbf{s}_j^{(1)} - \mathbf{d}_j\right). \tag{25}$$

The above equation can be written for each of the six actuators ($j = 1, \ldots, 6$). The obtained set of six equations should be solved for the unknown quantities $\mathbf{r}$ and $\mathbf{R}$. Six scalar equations (25) can be rewritten as one vector equation:

$$\boldsymbol{\Phi}(\mathbf{q}) \equiv \begin{bmatrix}\Phi_1(\mathbf{q}) & \cdots & \Phi_6(\mathbf{q})\end{bmatrix}^T = \mathbf{0}_{6\times1}, \tag{26}$$

with Φ_j defined as:

$$\Phi_j(\mathbf{q}) \equiv \Phi_j(\mathbf{r}, \boldsymbol{\varphi}) \equiv \left(\mathbf{r} + \mathbf{R}(\boldsymbol{\varphi})\mathbf{s}_j^{(1)} - \mathbf{d}_j\right)^T\left(\mathbf{r} + \mathbf{R}(\boldsymbol{\varphi})\mathbf{s}_j^{(1)} - \mathbf{d}_j\right) - l_j^2 = 0 . \tag{27}$$

The above set of six nonlinear algebraic equations will be solved using an iterative Newton-Raphson method. Several solutions can be found, however we are interested only in this one, which corresponds to the admissible configuration of the manipulator. That is why the initial guess $\mathbf{q}^0$ should be chosen carefully. It was found, that good results are obtained, when the iterations start from point $\mathbf{q}^0$, which represents the central point of the manipulator workspace. Some numerical tests have proven that iterations are converging to the proper solution. The Newton-Raphson iteration process uses the following formula:

$$\mathbf{q}^{k+1} = \mathbf{q}^k - \left[\boldsymbol{\Phi}_{\mathbf{q}}\left(\mathbf{q}^k\right)\right]^{-1}\boldsymbol{\Phi}\left(\mathbf{q}^k\right). \tag{28}$$

The Newton-Raphson method requires vector function (26) to be differentiated with respect to the vector of unknown $\mathbf{q}$ coordinates. Let us start with partial derivatives of rotation matrix (1):

$$\left(\mathbf{R}(\boldsymbol{\varphi})\right)_{\varphi_1} = \boldsymbol{\Omega}_z \mathbf{R}_z(\varphi_1) \mathbf{R}_x(\varphi_2) \mathbf{R}_z(\varphi_3),$$
$$\left(\mathbf{R}(\boldsymbol{\varphi})\right)_{\varphi_2} = \mathbf{R}_z(\varphi_1) \boldsymbol{\Omega}_x \mathbf{R}_x(\varphi_2) \mathbf{R}_z(\varphi_3), \tag{29}$$
$$\left(\mathbf{R}(\boldsymbol{\varphi})\right)_{\varphi_3} = \mathbf{R}_z(\varphi_1) \mathbf{R}_x(\varphi_2) \boldsymbol{\Omega}_z \mathbf{R}_z(\varphi_3),$$

where constant $\boldsymbol{\Omega}_x$ and $\boldsymbol{\Omega}_z$ matrices are defined as:

$$\boldsymbol{\Omega}_x \equiv \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{\Omega}_z \equiv \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{30}$$

We can use the above formulas when differentiating Eq. (27). Finally we obtain (for $j = 1, \dots , 6$):

$$\left(\Phi_j(\mathbf{r}, \boldsymbol{\varphi})\right)_{\mathbf{r}} \equiv 2\left(\mathbf{r} + \mathbf{R}(\boldsymbol{\varphi})\mathbf{s}_j^{(1)} - \mathbf{d}_j\right)^T \equiv 2\mathbf{l}_j^T, \tag{31}$$

$$\left(\Phi_j(\mathbf{r}, \boldsymbol{\varphi})\right)_{\varphi_k} \equiv 2\left(\mathbf{r} + \mathbf{R}(\boldsymbol{\varphi})\mathbf{s}_j^{(1)} - \mathbf{d}_j\right)^T \left(\mathbf{R}(\boldsymbol{\varphi})\right)_{\varphi_k} \mathbf{s}_j^{(1)} \equiv 2\mathbf{l}_j^T \left(\mathbf{R}(\boldsymbol{\varphi})\right)_{\varphi_k} \mathbf{s}_j^{(1)}, \quad k = 1, 2, 3. \tag{32}$$

Let us recapitulate the position problem. The equations to be solved are given by formulas (27). Matrix $\boldsymbol{\Phi}_\mathbf{q}$ elements can be calculated using Eq. (31) and Eq. (32). The iterations are described by (28).

After solving the position problem, vector $\mathbf{r}$ and matrix $\mathbf{R}$ are known. Equations (6), (7), (8) and (17) (for $j = 1, \dots , 6$) can be used to calculate the manipulator Jacobian matrix $\mathbf{J}$. Six scalar equations (16) can be rewritten as one vector equation:

$$\dot{\mathbf{L}} = \begin{bmatrix} \dot{l}_1 \\ \vdots \\ \dot{l}_6 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_6 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}. \tag{33}$$

In the direct problem actuator velocities $\dot{l}_j$ are given. The unknown platform velocities $\mathbf{v}$ and $\boldsymbol{\omega}$ one can found solving linear equations set (33).

After solving the position and velocity problems, quantities $\mathbf{r}$, $\mathbf{R}$, $\mathbf{v}$ and $\boldsymbol{\omega}$ are known. Equations (10) and (21) (for $j = 1, \dots , 6$) can be utilized to calculate $\dot{\mathbf{l}}_j$ and $\dot{\mathbf{u}}_j$. Scalar equations (22), for $j = 1, \dots , 6$, can be combined into one vector equation:

$$\mathbf{J} \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\varepsilon} \end{bmatrix} = \begin{bmatrix} \ddot{l}_1 - \dot{\mathbf{u}}_1^T \mathbf{l}_1 - \mathbf{u}_1^T \tilde{\boldsymbol{\omega}} \tilde{\boldsymbol{\omega}} \mathbf{s}_1 \\ \vdots \\ \ddot{l}_6 - \dot{\mathbf{u}}_6^T \mathbf{l}_6 - \mathbf{u}_6^T \tilde{\boldsymbol{\omega}} \tilde{\boldsymbol{\omega}} \mathbf{s}_6 \end{bmatrix}. \tag{34}$$

The actuator accelerations $\ddot{l}_j$ are known in the direct problem, hence unknown platform linear and angular accelerations ($\mathbf{a}$ and $\boldsymbol{\varepsilon}$) can be calculated from linear equations set (34).

3. Manipulator dynamics

3.1. Forward dynamics

The forward dynamics problem consists in searching for the mechanism motion, when forces articulating the mechanism are known. The forward dynamics of manipulator has been modelled using multibody package. This program automatically generates and solves multibody system equations of motion. Thus, there is no need to derive the motion equations in a full (i.e. not simplified) form.

3.2. Inverse dynamics (simplified)

The inverse problem of dynamics consists in searching for driving forces, which are necessary to obtain the requested motion of mechanism. The manipulator control system employs the simplified inverse dynamics model. To simplify calculations, it was assumed that all the parts of mechanism, except for the moving platform, are massless. Moreover, friction in joints was neglected (the only exception was friction in the hydraulic actuators, which is described in section 0).

It was assumed that platform centre of mass coincides with the origin of π_1 frame. The platform is characterized by mass m and inertia matrix $\mathbf{I}^{(1)}$. The matrix $\mathbf{I}^{(1)}$ elements are calculated with respect to the local (moving with the platform) frame π_1, thus they are constant. The inertia properties calculated with respect to the centre of mass and axes parallel to the global frame π_0 are not constant and depend on the platform instantaneous orientation. The inertia matrix (with respect to axes parallel to π_0 frame) can be calculated using the following equation:
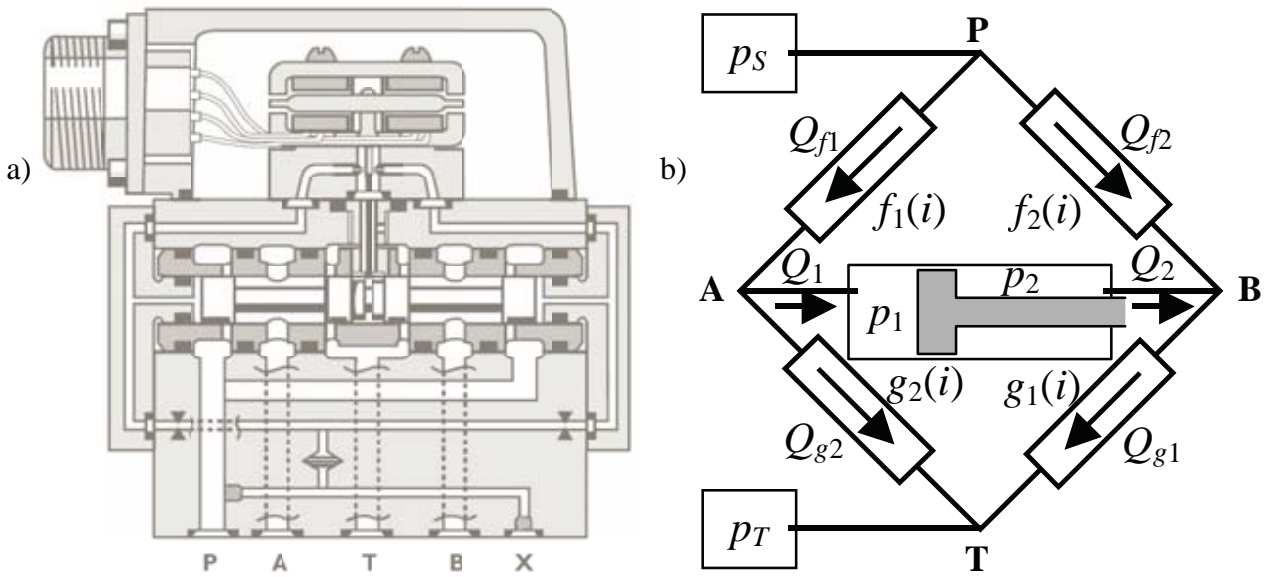

Fig. 3. a) A drawing of a real servovalve, b) Schematic model of servovalve

$$\mathbf{I} = \mathbf{R}\,\mathbf{I}^{(1)}\mathbf{R}^{T}. \tag{35}$$

The Newton law relates the total force acting on the platform with the platform mass and centre of mass linear acceleration:

$$\mathbf{F} = m\mathbf{a}. \tag{36}$$

The Euler equation relates the total torque about the platform centre of mass with the platform angular velocity, acceleration and inertia matrix:

$$\mathbf{M} = \mathbf{I}\boldsymbol{\varepsilon} + \tilde{\boldsymbol{\omega}}\,\mathbf{I}\,\boldsymbol{\omega}. \tag{37}$$

For assumed platform motion, force $\mathbf{F}$ and torque $\mathbf{M}$ can be calculated directly from equations (36) and (37).

The manipulator Jacobian matrix $\mathbf{J}$ relates forces P_j developed by actuators with total force $\mathbf{F}$ and total torque $\mathbf{M}$ [1, 8]:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} = -\mathbf{J}^{T}\mathbf{P}, \quad \mathbf{P} = \begin{bmatrix} P_1 & \dots & P_6 \end{bmatrix}^{T}. \tag{38}$$

To solve the inverse problem of dynamics it is sufficient to perform calculations according to Eq. (35) – (37) and then to solve the linear set of equations (38).

3.3. Friction in actuators

The actuator output P_j force differs form resultant of hydraulics P_j^H force which is applied to the piston. The difference is caused by the presence of friction effects in actuator. The following model of *j*-th actuator friction force P_j^F has been used by the control package during simulations:

$$P_j^F = \begin{cases} F_C\,\mathrm{sgn}(\dot{l}_j) + b\,\dot{l}_j & \dot{l}_j \neq 0 \\ P_j^{ext} & \dot{l}_j = 0, \left|P_j^{ext}\right| < F_S \\ F_S\,\mathrm{sgn}(P_j^{ext}) & \dot{l}_j = 0, \left|P_j^{ext}\right| \geq F_S \end{cases}, \tag{39}$$

where b is the viscous friction coefficient, F_C is the Coulomb friction force, F_S is the maximal stiction force, and P_j^{ext} is the external force.

4. Electrohydraulic actuator

4.1. Governing equations

The equations describing actuator with electrohydraulic servovalve presented in [4] and [5], with some modifications however, were utilized in this study. Control of a hydraulic system is achieved through the use of servovalves, see Fig. 3(a). Only the resistive effect of a valve is considered here, since their natural frequency is much higher than that of the mechanical load. It is also assumed that the geometry of the valve is ideal, e.g. the valve has sharp edges and zero cross leakage, [2, 7].

A typical hydraulic servovalve consists of four symmetric and matched servovalve orifices making up flow paths through four nonlinear resistors, modulated by the input voltage, see Fig. 3(a). Thereby, the servovalve is modelled as the hydraulic equivalent of a Wheatstone bridge, see Fig. 3(b). When the servovalve input current is positive, $i > 0$, flow passes through the orifices 1 and 3 (path P – A – B – T), and flow leakages exist in the valve orifices 2 and 4. Similarly, when the servovalve input current is negative, $i < 0$, flow passes through the path P – B – A – T, and flow leakages exist in the valve orifices 1 and 3. This model is described by:

$$\begin{aligned} Q_{f1} &= f_1(i,\,C_d,\,\rho)\sqrt{p_S - p_1}, & Q_{g1} &= g_1(i,\,C_d,\,\rho)\sqrt{p_2 - p_T}, \\ Q_{f2} &= f_2(i,\,C_d,\,\rho)\sqrt{p_S - p_2}, & Q_{g2} &= g_2(i,\,C_d,\,\rho)\sqrt{p_1 - p_T}, \end{aligned} \tag{40}$$

where Q_{f1}, Q_{f2}, Q_{g1} and Q_{g2} are the servovalve flows through the orifices 1, 2, 3 and 4 respectively, p_S and p_T are the power supply and return pressure of the servosystem, correspondingly, i is the servovalve motor current (control command), and $f_1(i, C_d, \rho)$, $f_2(i, C_d, \rho)$, $g_1(i, C_d, \rho)$ and $g_2(i, C_d, \rho)$ are nonlinear functions in the servovalve motor current, the discharge coefficient C_d and the mass density of the fluid, ρ. In general, the discharge coefficient is as function of the *Reynolds* number and valve geometry. However, fluid density and Reynolds dependencies are weak for turbulent flow and therefore only the current dependency is significant here, [5]; therefore, the functions $f_1(i, C_d, \rho)$, $f_2(i, C_d, \rho)$, $g_1(i, C_d, \rho)$ and $g_2(i, C_d, \rho)$ are reduced to $f_1(i)$, $f_2(i)$, $g_1(i)$ and $g_2(i)$, correspondingly. Because of servovalve symmetry, the current functions are given by:

$$\begin{aligned} f_1(i) &= g_1(i) = f_2(-i) = g_2(-i), \\ f_2(i) &= g_2(i) = f_1(-i) = g_1(-i). \end{aligned} \tag{41}$$

Experimental results [3] showed that it is a good approximation to assume that these functions are linear functions of the input current, when flow passes through the main path, and have a constant value when flow passes through the leakage flow path. For instance when $i > 0$, the main flow path passes through the orifices 1 and 3 and therefore the functions of Eq. (40) are written as:

$$f_1(i) = g_1(i) = K_0 + K_1 \cdot i,$$
$$f_2(i) = g_2(i) = K_0,$$
(42)

where K_1 and K_0 are positive constants, which correspond to the main and leakage valve flow paths. In the above equations constant coefficient K_1 corresponds to the main valve flow path and constant coefficient K_0 corresponds to the leakage valve flow path. Due to symmetry of the valve, the coefficients K_1 and K_0 are the same for all the flow paths.

The flow through the piston side chamber port of hydraulic cylinder (Q_1) and the flow through the rod side chamber port (Q_2) can be calculated as:

$$Q_1 = Q_{f1} - Q_{g2},$$
$$Q_2 = Q_{g1} - Q_{f2}.$$
(43)

The flows Q_1 and Q_2 depend on $\dot{l}$ – the velocity of the piston with respect to the cylinder:

$$Q_1 = A_1 \dot{l}, \qquad Q_2 = A_2 \dot{l},$$
(44)

where A_1 is the piston side area and A_2 is the rod side area.

The resultant hydraulics piston force can be calculated as:

$$P^H = p_1 A_1 - p_2 A_2.$$
(45)

4.2. Actuator force calculation

During the manipulator motion simulation it is necessary to calculate the actuators forces. The force generated by actuator depends on two factors: the control current and the actuator velocity.

Substituting Eq. (44) and (40) into Eq. (43) yields:

$$f_1\sqrt{p_S - p_1} - g_2\sqrt{p_1 - p_T} - A_1 \dot{l} = 0,$$
(46)

$$g_1\sqrt{p_2 - p_T} - f_2\sqrt{p_S - p_2} - A_2 \dot{l} = 0.$$
(47)

The first of the above equations enables to calculate pressure p_1, and the second pressure p_2. Equation (46) can be raised two times to the power two, to obtain a quadratic equation in p_1. Solving the quadratic equation results in the following (the solution belonging to the interval $[p_T, p_S]$ is the only being considered):

$$p_1 = \frac{(p_T + p_S)f_1^2 g_2^2 + p_T g_2^4 + p_S f_1^4 - (f_1^2 - g_2^2)A_1^2 \dot{l}^2 \mp 2 f_1 g_2 A_1 \dot{l}\sqrt{(p_S - p_T)(f_1^2 + g_2^2) - A_1^2 \dot{l}^2}}{(f_1^2 + g_2^2)^2}.$$
(48)

Solving equation (47) for p_2 yields:

$$p_2 = \frac{(p_T + p_S)f_2^2 g_1^2 + p_T g_1^4 + p_S f_2^4 - (f_2^2 - g_1^2)A_2^2 \dot{l}^2 \pm 2 f_2 g_1 A_2 \dot{l}\sqrt{(p_S - p_T)(f_2^2 + g_1^2) - A_2^2 \dot{l}^2}}{(f_1^2 + g_2^2)^2}.$$
(49)

It is worth noting that equations (46) and (47) have been two times raised to the power two, to obtain quadratic equations. Thus, it can happen that pressures p_1 and p_2 fulfil the appropriate quadratic equations, but do not fulfil the original equations (46) and (47). That is why it is necessary to check whether the pressures p_1 and p_2 obtained from Eq. (48) and (49) fulfil Eq. (46) and (47).

Having pressures p_1 and p_2 calculated, the force can be found directly from Eq. (45).

The equations presented in this section enable us to calculate actuator force as a function of control current and actuator velocity. This characteristic is presented in Fig. 4. During calculations the pressures have been confined to the interval $[p_T, p_S]$.

4.3. Control current calculation

The control system solves the inverse dynamics problem and determines the forces required to perform desired motion. Then, for given actuator velocity and required actuator force, appropriate control current must be calculated.

The dependency between required force P^H and control current i could be found in analytical form by substituting Eq. (48) and (49) into Eq. (45), and then utilizing Eq. (42). Unfortunately, the obtained equation would be too complicated to solve it analytically. Therefore a numerical method was used.
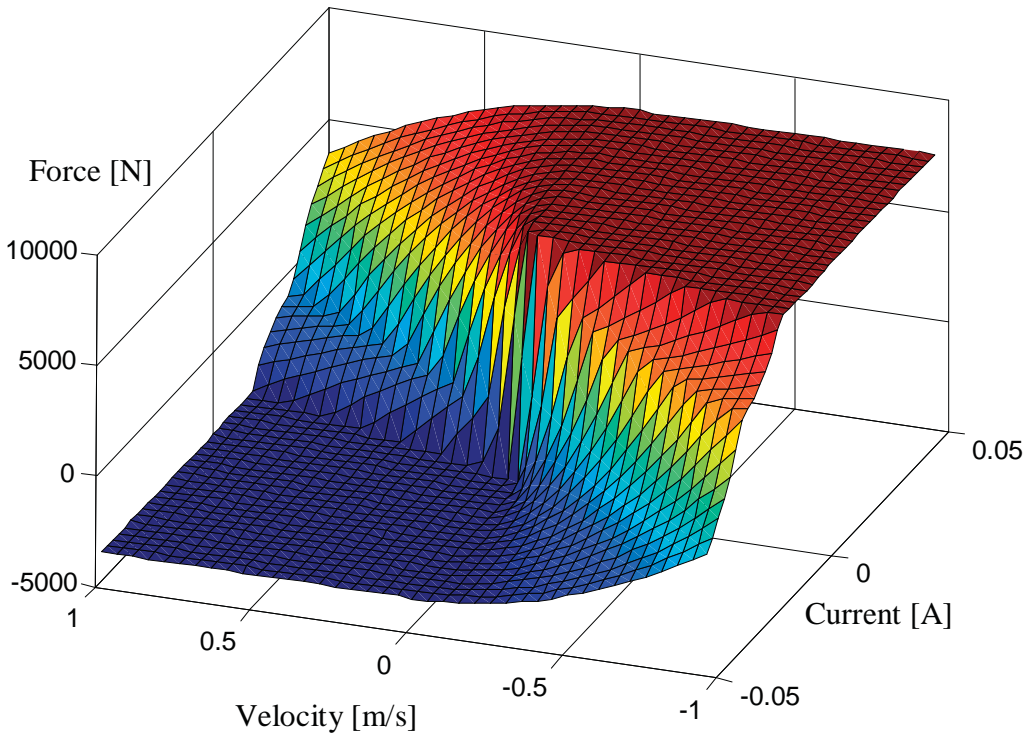

Fig. 4. Actuator force as a function of control current and velocity

To find the unknown control current i a bisection method was used. The procedure calculating force P^H, for given current i and velocity $\dot{l}$ was utilized (see previous section). In our problem velocity $\dot{l}$ is given, thus function $P^H(i, \dot{l})$ can be treated as a function of one variable, namely current i. The unknown control current must belong to the interval $[-i_{max}, i_{max}]$. The plot in Fig. 4 shows, that for selected value of $\dot{l}$, the force P^H is a monotonic function of current i That is why it is possible to use the bisection method.

The bisection procedure works properly, if for required force P_D^H and given velocity $\dot{l}$ the following condition is fulfilled:

$$P^H(i_{min}, \dot{l}) \quad < \quad P_D^H \quad < \quad P^H(i_{max}, \dot{l}). \tag{50}$$

In the other case, the required force is out of admissible range. As a result of calculations we obtain control current equal to i_{max} (or $-i_{max}$), which corresponds to maximal (or minimal) available force.

5. Control system

A novel model-based controller for six-dof electrohydraulic Stewart platforms is developed [4]. Desired Cartesian trajectories yield the desired actuator trajectories using mechanism inverse kinematics. The control law provides the current sent to the linear hydraulic servoactuator servovalves, so that the error dynamics converge asymptotically to zero, independent of load variations. The developed control analysis is based on the system dynamic and hydraulic models; therefore, it is assumed that the dynamic terms of the system are known. In this approach, force, pressure or acceleration feedback is not required.

The control law is designed to reduce the control errors on position and velocity levels simultaneously. The control currents are calculated to satisfy the following error dynamics equation:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0}, \tag{51}$$

where $\mathbf{e} = \mathbf{L}_D - \mathbf{L}$ is the position error ($\mathbf{L}_D$ is a 6-element vector of desired actuator lengths), $\mathbf{K}_p = k_p \mathbf{I}_{6\times 6}$ and $\mathbf{K}_v = k_v \mathbf{I}_{6\times 6}$ are diagonal matrices of control gains. The gain coefficients k_p and k_v are selected to achieve the critical damping of the system described by equation (51).

At the beginning of computation the required motion of the platform is calculated. Then the inverse kinematics problem is solved to find the actuator desired lengths of $\mathbf{L}_D$, velocities $\dot{\mathbf{L}}_D$ and accelerations $\ddot{\mathbf{L}}_D$.

In the real manipulator the actual actuators lengths $\mathbf{L}$ and velocities $\dot{\mathbf{L}}$ are measured by appropriate sensors. In the simulational model these values are computed by the multibody package, to provide feedback for the control system model.

The next step of computations consists in accelerations calculation. For given vectors of $\mathbf{L}_D$, $\dot{\mathbf{L}}_D$, $\ddot{\mathbf{L}}_D$, $\mathbf{L}$ and $\dot{\mathbf{L}}$, the vector of accelerations $\ddot{\mathbf{L}}$, which satisfies equation (51), is calculated:

$$\ddot{\mathbf{L}} = \ddot{\mathbf{L}}_D + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \ddot{\mathbf{L}}_D + \mathbf{K}_v \left(\dot{\mathbf{L}}_D - \dot{\mathbf{L}} \right) + \mathbf{K}_p \left(\mathbf{L}_D - \mathbf{L} \right). \tag{52}$$

Then the inverse problem of dynamics is solved. The driving forces necessary to produce the required motion (described by $\mathbf{L}$, $\dot{\mathbf{L}}$ and $\ddot{\mathbf{L}}$) are calculated. Friction forces in actuators are taken into account. The inverse dynamics calculations must be preceded by direct kinematics solution, to obtain the platform motion. The last step of computations consists in calculation of control currents for all actuators.

6. Simulation results

Schematic view of the simulational model is presented in Fig. 1. Several simulations were performed in order to check if it is the impact of the inverse dynamic model simplifications on the quality of control process.

The same platform desired trajectory has been used in all the simulations. The moving platform desired Cartesian trajectory was described by the following equations:

$$\mathbf{r}(t) = \begin{bmatrix} x\sin(2\pi f \cdot t) \\ y\cos(2\pi f \cdot t) \\ z_1 + z\sin(2\pi f \cdot t) \end{bmatrix}, \qquad \begin{aligned} \varphi_1(t) &= \alpha\cos(2\pi f \cdot t), \\ \varphi_2(t) &= \pi/2 + \beta\sin(2\pi f \cdot t), \\ \varphi_3(t) &= \gamma\sin(2\pi f \cdot t), \end{aligned} \tag{53}$$

where parameters $x, y, z, z_1, \alpha, \beta, \gamma, f$ are constant values and time $t \in [0 \quad \tau]$.

The manipulator in different stages of motion is presented in Fig. 5.

The quality of control process assessment was based on the position error $\mathbf{e} = \mathbf{L}_D - \mathbf{L}$ (difference between desired and obtained actuator lengths) and velocity error $\dot{\mathbf{e}} = \dot{\mathbf{L}}_D - \dot{\mathbf{L}}$. The

control errors **e** and $\dot{\mathbf{e}}$ are time-varying six-element vectors. In order to make the simulation comparison less difficult, the following mean errors have been defined:

$$e_p = \sqrt{\frac{1}{\tau} \int_0^\tau \overline{\mathbf{e}(t)^T \mathbf{e}(t)} dt} \,, \qquad e_v = \sqrt{\frac{1}{\tau} \int_0^\tau \overline{\dot{\mathbf{e}}(t)^T \dot{\mathbf{e}}(t)} dt} \,. \tag{54}$$

Several variants of simulations were performed. The simulations descriptions and concise results, i.e. mean errors e_p are e_v presented in table 1.
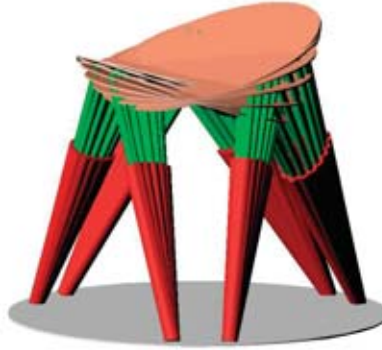
Fig. 5. Superimposed view of platform in motion

Tab. 1. Simulation characteristics and results

	Control system inverse dynamics (control software)	Simulational model (multibody software)	e_p [mm]	e_v [mm/s]
A.	Friction neglected	Friction neglected Actuator masses neglected	0.18	0.68
B.	Friction neglected	Friction neglected Actuator masses included	3.50	4.03
C.	Friction neglected	Friction included Actuator masses neglected	7.77	33.11
D.	Friction neglected	Friction included Actuator masses included	8.52	31.92
E.	Friction included	Friction included Actuator masses included	3.50	4.19
F.	Friction included Platform mass corrected	Friction included Actuator masses included	0.65	2.15
G.	Friction neglected Platform mass corrected	Friction included Actuator masses included	7.70	31.73
H.	Friction underestimated Platform mass corrected	Friction included Actuator masses included	4.04	19.14
I.	Friction overestimated Platform mass corrected	Friction included Actuator masses included	3.89	14.68
J.	Friction included Platform mass corrected	Friction included Actuator masses included Geometrical parameters changed by 1%	1.33	6.20
K.	Friction included Platform mass corrected	Friction included Actuator masses included Payload attached to the platform	10.80	14.56
L.	Friction included Platform mass corrected	Friction included Actuator masses included Payload flexibly attached	6.30	11.02
M.	Friction included Platform mass corrected k_p and k_v enlarged	Friction included Actuator masses included Payload flexibly attached	1.62	2.94

To enable proper interpretation of the data presented in table 1, some important issues must be emphasized:

All simulations were performed for the same control gains $k_p = 64\pi^2$ and $k_v = 16\pi$. The only exception was simulation L, for which the gains were greater: $k_p = 256\pi^2$, $k_v = 32\pi$.

The actuator masses were neglected in the inverse dynamics model. The moving platform was the only system element with non-zero mass. In majority of the simulations, the multibody model performing the direct dynamics calculations was not neglecting the masses of actuators. Only during simulations A and C the actuator masses were set to zero.

In the inverse dynamics model, which is employed by the control system, the actuator masses may be considered in the simplified way, by appropriate enlargement of the moving platform mass. The platform mass correction was done in simulations F-M.

The friction forces can be neglected or taken into account both in the multibody package direct dynamics model (used to simulate manipulator motion) and in the control software inverse dynamics model (utilized by the model-based control system). The friction parameters are difficult to measure, and moreover, they can be unstable. This was the reason to perform two simulations, during which the friction parameters used by the inverse dynamics model were different than friction parameters used by the direct dynamics model. Simulation H was performed for underestimated (by 50%) friction parameters, and simulation I was performed for overestimated (by 50%) friction parameters.

The geometrical parameters of platform model used by the control system can differ from the real platform parameters. Simulation J was performed to investigate the effects of erroneous parameter estimation, the platform parameters differed by 1% from those used in the controller.

The manipulator in three simulations has carried a payload. In the case of simulation K a 50 kg mass was rigidly attached to the moving platform. In the cases of L and M simulations, a 30 kg mass was attached to the moving platform via spherical joint and a system of springs and dampers.

The table 1 presents only concise information about the simulations and the obtained control quality. Some interesting results are discussed below in a more detailed way.
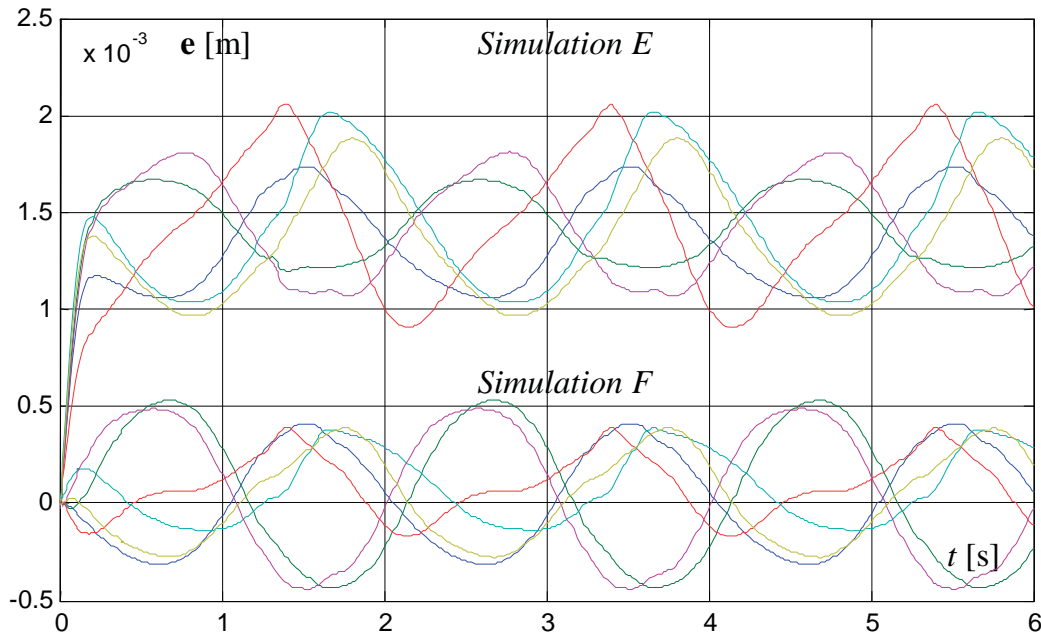
Simulation A refers to the situation in which the inverse dynamics model fully corresponds to the forward dynamics model. The obtained control errors are almost zero. During simulation B the actuators (in the multibody package model) were not massless anymore. Thus the inverse dynamics model utilized for model-based control was simplified and did not fully correspond with the manipulator dynamics. During simulation C actuator friction force was neglected in the inverse dynamics model, but was taken into account in the direct dynamics calculations. In other words, the model based control neglected friction effects, which were present in the manipulator. The results of simulations A, B and C comparison leads to the statement, that neglecting the friction effects causes much bigger control problems, than neglecting the actuator masses in the inverse dynamics calculations.

The problems caused by neglecting actuator masses in the inverse dynamics model can be partially neutralized by making appropriate enlargement of the moving platform mass. The position error during simulation E is presented in Fig. 6. In this simulation the platform mass in the inverse dynamics model remained unchanged. It is worth noting, that the position errors oscillate around a mean value of approximately 1.5 mm. In the same figure the results of simulation F are presented. During this simulation the platform mass in the inverse dynamics model was enlarged by sum of the piston-side masses of the actuators. It is clearly visible that this time the position errors oscillate around a close-to-zero value. The error oscillation amplitudes obtained in both simulations have similar magnitudes.

Employing a friction model in the model-based control inverse dynamics calculations can reduce the problems caused by actuators friction. It should be stated, however, that friction is a complicated phenomenon and its computational models are usually severely simplified. Moreover, the friction parameters are usually difficult to measure and quite often are time-varying. Thus, it

should be expected that the friction model utilized during the inverse dynamics calculations would not be accurate. A series of simulations was preformed in order to check if it is the influence of friction forces and friction model inaccuracies on the obtained quality of control.

Fig. 6. Simulations E and F: position errors vs. time

Simulation F refers to the situation, in which the friction model used for the inverse dynamics calculation fully corresponds to the friction effects in manipulator (modelled in multibody package). During simulation G the friction effects were neglected in the inverse dynamics calculations (but still present in the manipulator). The results of friction neglecting are clearly visible as "peaks" on the velocity error diagram in Fig. 7. Friction-stiction transition effects at actuators velocities close to zero cause the biggest problems. Simulation H shows what happens, when friction model is utilized by the control system, but friction parameters are underestimated (reduced by half). Comparing with simulation G, the improvement of control quality is visible, however the problems with stiction are still present. During simulation I the friction parameters have been overestimated (enlarged by 50%). The results show, that the velocity errors are greater than observed during simulation F, but significantly lesser than observed during simulation G.

Simulation J was performed to investigate the effects of erroneous estimation of the platform geometrical parameters. During this simulation the multibody model parameters differed by 1% from those used in the controller. It was found that relatively small changes of the geometrical parameters led to relatively big control errors. The error estimates presented in the table 1 are based on the actuators' position and velocity errors. It should be noted, that in the case of simulation J, the most significant are the position and velocity errors in the Cartesian space. In the case of erroneous geometrical parameters, accurate actuators motion does not result in accurate platform motion.

The inverse dynamics model employed by the model-based control system is tuned to an average mass of payload. The changes of the payload mass or external force application are treated as the control disturbances. Simulations K and L have been performed in order to check the influence of payload carrying on the quality of control. In the case of simulation K a 50 kg cylinder was rigidly fixed to the moving platform. In the case of simulation L a 30 kg inverted pendulum of length 0.2 m was attached to the platform via spherical joint. The pendulum is supported by a system of springs and dampers. The flexible mounting of pendulum enabled testing the manipulator subjected to non-constant loads.
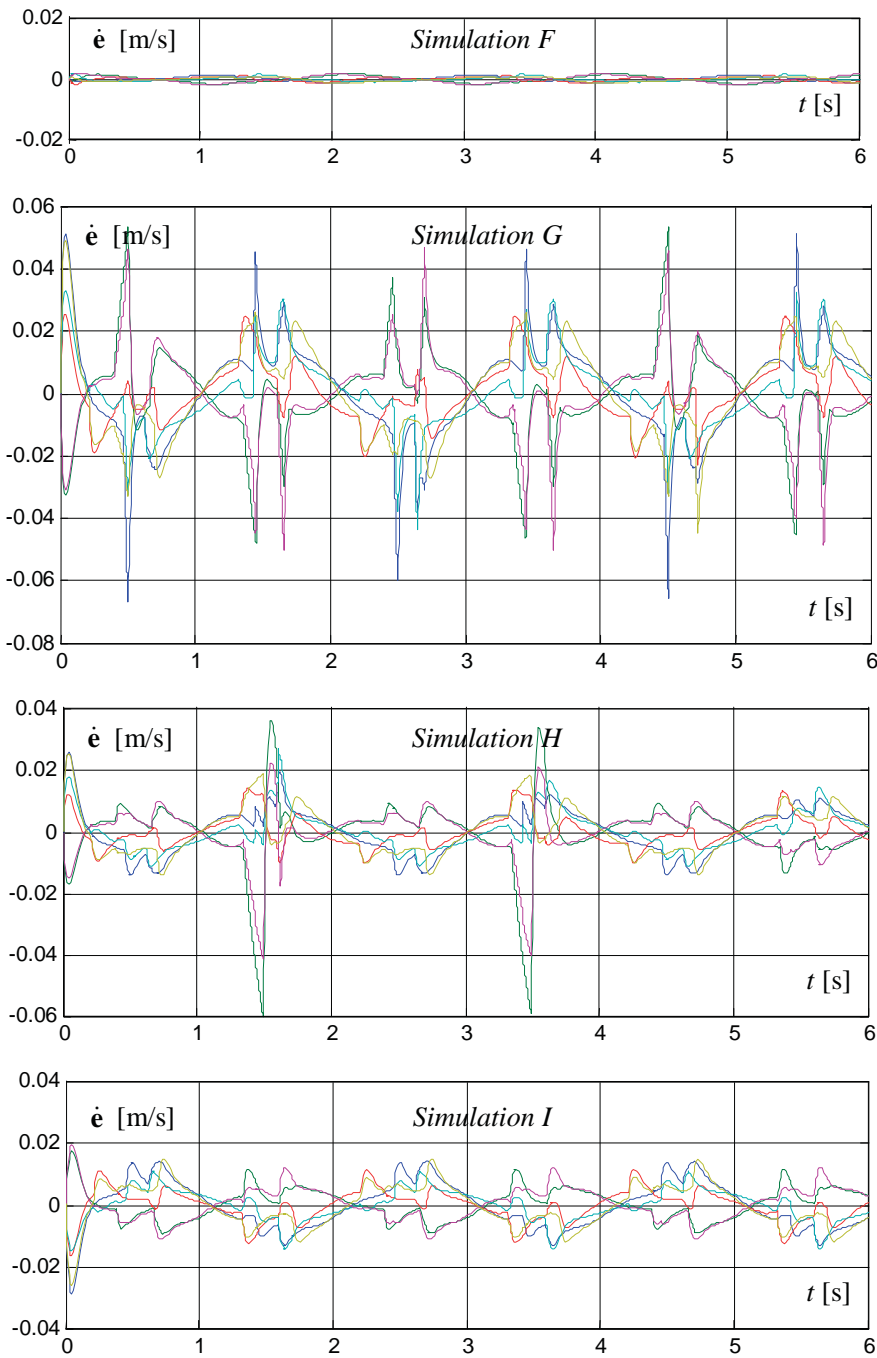
Fig. 7. Simulations F, G, H and I: velocity errors vs. time

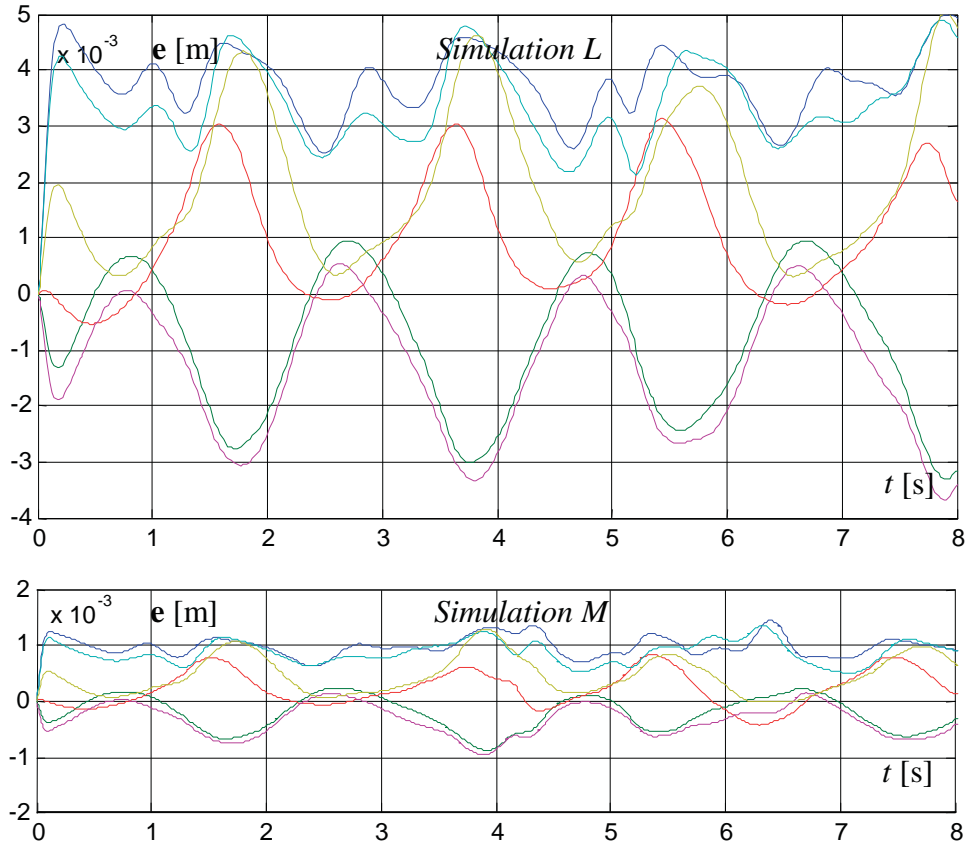*Fig. 8. Models used during simulations K (left), L and M (right)*

Fig. 9. Simulations L and M: position errors vs. time

The results of simulation L are presented in Fig. 9. It is visible, that the position errors do not stabilize to be close to periodical functions (as it was observed in the other simulations). This is a result of changes in external loads, caused by the pendulum motion. Moreover the position errors are greater than errors observed during simulation F (Fig. 6). It should be pointed out that additional masses of 50 kg or 30 kg are relatively big in comparison with platform mass (300 kg). In consequence the observed control errors are significant as well.

All the discussed earlier simulations were characterized by the following control gains: $k_p = 64\pi^2$ and $k_v = 16\pi$. The proper choice of gains k_p and k_v is crucial for the quality of control. The k_p and k_v gains influence on the control errors is usually greater than the influence of inverse dynamics model simplifications and other factors discussed above. Simulation M was performed to show the importance of control gains k_p and k_v. The only difference between simulations L and M is, that in the second case the control gains were the following: $k_p = 256\pi^2$ and $k_v = 32\pi$. The obtained position errors were significantly smaller in the case of simulation L, as it is shown in Fig. 9.

7. Concluding remarks

The presented simulational model of parallel manipulator with electrohydraulic actuators and model-based control system enables to analyze various problems concerning the system behaviour. The presented study was focused on checking if it is the influence of dynamics model simplifications and model parameter uncertainties on the quality of control process.

The obtained results have shown, that inverse dynamics model simplifications consisting in neglecting the mass of actuators have relatively little influence on position errors and even smaller influence on velocity errors. It was also found, that the effects of actuators masses neglecting can be reduced by appropriate changes in the modelled mass of the platform.

It was found that friction effects should be introduced to the inverse dynamics model, since it importantly improves the quality of control. The simulation results have shown, that the friction parameters should be identified with possibly big accuracy. If the friction model parameters are not accurate enough, the quality of control does not improve. It was also found during investigations, that the parameters describing stiction-friction transition effects are the crucial ones.

Note that the developed presented model was run for off-line simulations. However selected procedures (namely: direct kinematics, inverse dynamics and control currents computations), can be implemented in a controller running under a real-time environment. This is due to the fact that the computation time of the direct kinematics, inverse dynamics and control procedure is estimated to be between 5 to 15 ms, i.e. small enough to satisfy the requirements for a real-time mechanical system.

At the end it is worth noting, that the presented simulational model can be easily modified, thus it can be used to model the manipulator interactions with the environment.

References

[1] Angeles, J., *Fundamentals of Robotic Mechanical Systems,* Springer Science+Business Media, 3rd Edition, 2007.

[2] Blackburn, J. F., Reethof, G., Shearer, J. L., *Fluid Power Control*, Cambridge, MA: MIT Press, 1960.

[3] Davliakos, I., Zafiris, A., Papadopoulos, E., *Joint Space Controller Design for Electrohydraulic Servos,* Proc. 2006 IEEE International Symposium on Computer-Aided Control Systems Design, (CACSD '06), pp. 796-801, 2006.

[4] Davliakos, I., Papadopoulos, E., *Invariant Error Dynamics Controller for a 6-dof Electrohydraulic Stewart Platform,* Proc. 6th CISM-IFToMM Symposium on Robot Design, Dynamics and Control, (ROMANSY '06), Warsaw 2006.

[5] Davliakos, I., Chatzakos, P., Papadopoulos, E**.,** *Development of a Model-based Impedance Controller for Electrohydraulic Servos*, Proc. International Conference on Robotics and Applications, Cambridge, MA, USA 2005.

[6] Merritt, H. E., *Hydraulic Control Systems,* J. Wiley, 1967.

[7] Thayer, W.J., *Specification Standards for Electrohydraulic Flow Control Servovalves,* Technical Bulletin 117, Moog Incorporation Control Division, E. Aurora, New York 1962.

[8] Tsai, L.-W., *Robot Analysis, The Mechanics of Serial and Parallel Manipulators,* John Wiley & Sons Inc., New York 1999.

Acknowledgements