Dynamic analysis and control of robotic manipulator for chemically aggressive environments

Martin Goubej
Department of Cybernetics
University of West Bohemia
Pilsen, Czech Republic
Email: mgoubej@kky.zcu.cz

Martin Švejda
Department of Cybernetics
University of West Bohemia
Pilsen, Czech Republic
Email: msvejda@kky.zcu.cz

***Abstract*—The paper deals with newly developed industrial robotic manipulator with special serio-parallel architecture which was designed for operation in chemically aggressive environment. Dynamical analysis of the manipulator is performed for the purpose of control law synthesis. General method for derivation of static models is presented.**

***Index Terms*—Industrial robots, inverse and direct kinematic problem, force balance, decentralized PID control.**

I. Introduction

Utilization of industrial robots in modern automated factories has grown dramatically in recent decades. There is variety of industrial robots available on the market which can be well suited for standard applications (welding, soldering, palletizing, material handling). However, the most commonly used universal robots may fail in case of some specific requirements and special design may be needed. One of such nonstandard applications is technology for robot supported parts cleaning which is essential in several fields of industry and mass production. A robot operates a degreasing or paint removal machine and manipulates with metal or nonmetal parts which have to be cleaned precisely. The goal is to remove all the remains of grease, cutting or tempering oil or any kind of mechanical dirt in order to prepare the parts for further processing. Procedure of cleaning, rinsing, drying and conservation in conjuction with strong chemicals such as acid, lye or special degreasing lotions is usually used to achieve high level of surface cleanness . These highly aggressive chemicals are dangerous not only for human staff but also for a robot which is used for manipulation and precise positioning of the parts inside a cleaning chamber. The sensitive parts of the robot such as drives, electronics or wiring can easilly be damaged and most of the commonly available robots are not suitable for this task. Therefore, serio-parallel manipulator which is called AGEBOT (AGgressive Environment roBOT) has been developed for specific operation in chemically aggressive environment.

AGEBOT is designed as a special robotic architecture which consist of two main parts - serial manipulator (SM) and parallel manipulator (PM), see [1]. SM ensures basic positioning of the end effector of AGEBOT including the translations in x, y, z axes and orientation of the longitudinal axis of PM. This motion is used for handling of parts which are

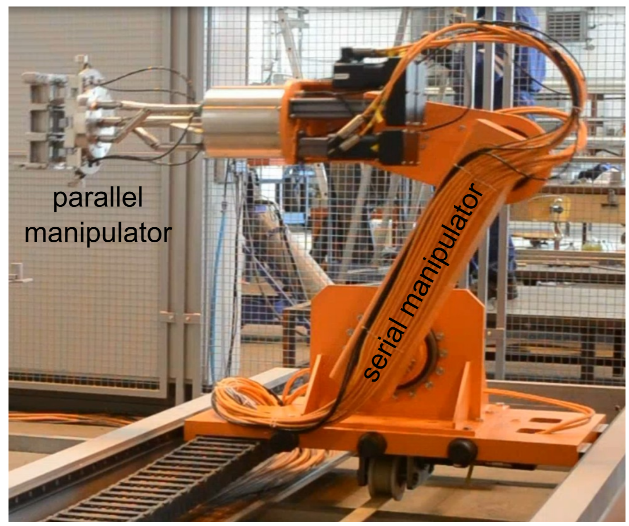

Fig. 1.   AGEBOT manipulator

to be processed in cleaning chambers. Parallel spherical wrist of the PM holds the end effector and performs positioning of the cleaned parts towards cleaning jets inside the chamber by changing their orientation in case of complex geometry of the parts. The main advantage of this kinematic structure is ability of waterproof separation of vulnerable components (motors, sensors, etc.) from an aggressive environment.

The main contribution of this paper is problem of AGEBOT manipulator control synthesis with respect to practical implementation issues. Many works related to robotics use centralized approach based on inverse dynamic model method and its modifications in order to obtain required force/torque setpoints to track desired end-effector trajectory, see [2], [3]. Such techniques are known for high computational burden and sensitivity to modelling errors. A question to be answered is whether it is necessary to use the complete dynamic model for control purpose in all cases or only the static one (gravity compensation) is sufficient and the computational load can be lowered. The comparison between dynamic and simplified static model is proposed. It is shown that general method for establishing the static model without necessity of explicit derivations of the position equations (forward and inverse kinematics) can be found. The comparison between dynamic and static modelling is presented.

The next part of the paper is related to application of standard hardware for motion control. Most of the commercially available servodrives are equipped with single axis cascade PID structure for position control. Development of special robot control hardware or significant changes in firmware of the drive may be needed in order to implement some more complex centralized control strategies. Our goal was to obtain simple tuning rules for decentralized PID control which is suitable for low-cost applications where the common industrial drives need to be used. Such choice of the control system structure may be appropriate in cases of less dynamical applications, where the static forces acting on the robot drives are prevalent and the dynamical interactions between the separate axes is relatively low. In consideration of this, the paper proposes a basic idea for tuning of cascade PID controller which is based on simplified dynamic model of AGEBOT.

A. *Serial part of AGEBOT*

SM consists of serial kinematic chain **PRRR** where all joints are actuated. The end effector of SM ensures 3 Dof (Degrees of Freedom), three translations and one rotation. The kinematic scheme is shown in Fig. 2 and joint $\boldsymbol{\Theta}_S$ and generalized $\boldsymbol{X}_S$ coordinates are set as:

$$\boldsymbol{\Theta}_S = \begin{bmatrix} d_1 & \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{1}$$

$$\boldsymbol{X}_S = \begin{bmatrix} \boldsymbol{O}_4^{0_S} \\ \phi \end{bmatrix} = \begin{bmatrix} x & y & z & \phi \end{bmatrix}^T \tag{2}$$

where $\boldsymbol{O}_4^{0_S}$ are the end effector coordinates with respect to coordinate system (CS) F_{0S}. The lengths of individual links are called the kinematic parameters of SM:

$$\boldsymbol{\xi}_S = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T$$

The dynamic parameters of SM are supposed to be link's masses $\boldsymbol{M}_S$, origins $\boldsymbol{CG}_S$ (with respect to Link's CSs) and inertia matrices $\boldsymbol{I}_S$ (with respect to Link's origins), m_e is a payload mass:

$$\boldsymbol{M}_S = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 & m_e \end{bmatrix}^T$$

$$\boldsymbol{CG}_S = \begin{bmatrix} \boldsymbol{cg}_1 & \boldsymbol{cg}_2 & \boldsymbol{cg}_3 & \boldsymbol{cg}_4 \end{bmatrix}^T$$
$$\boldsymbol{I}_S = \begin{bmatrix} \boldsymbol{I}_{S_1} & \boldsymbol{I}_{S_2} & \boldsymbol{I}_{S_3} & \boldsymbol{I}_{S_4} \end{bmatrix}$$

CSs of SM are established according to Denavit-Hartenberg notation (D-H), see [4]. Therefore, the homogeneous transformation matrices $\boldsymbol{T}_i^{i-1} \in \Re^{4\times4}$ which describe position and orientation of CS F_i with respect to CS F_{i-1} can be found. It is shown that the direct geometric model (DGM) $\boldsymbol{X}_S = \mathbf{F}_S(\boldsymbol{\Theta}_S, \boldsymbol{\xi}_S)$ and inverse geometric model (IGM) $\boldsymbol{\Theta}_S = \mathbf{F}_S^{-1}(\boldsymbol{X}_S, \boldsymbol{\xi}_S)$ can be solved in a closed form, for more details see [1].

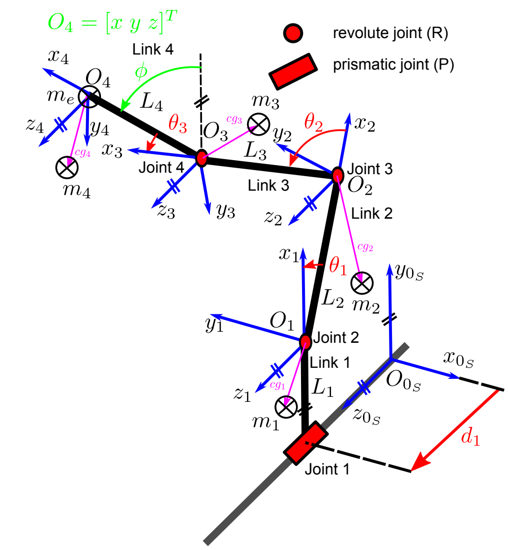

Fig. 2. Serial part of AGEBOT

B. *Parallel part of AGEBOT*

PM consists of 3 independent kinematic chains **PUS** where only **P** joints are actuated through joint coordinates $\boldsymbol{\Theta}_P$. A passive kinematic chain **S** reduces DoF of the end effector to 3 orientation DoF (XYZ Euler angles corresponding to generalized coordinates $\boldsymbol{X}_P$). The kinematic scheme is shown in Fig. 3. The kinematic and dynamic parameters are given in the same manner as for SM and they are identical for each kinematic chain.

$$\boldsymbol{\Theta}_P = \begin{bmatrix} l_{11} & l_{21} & l_{31} \end{bmatrix}^T, \ \boldsymbol{X}_P = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T \tag{3}$$

$$\boldsymbol{\xi}_P = \begin{bmatrix} a_1 & a_2 & l & v \end{bmatrix}^T \tag{4}$$

where a_1, a_2 is a side length of the base and end effector triangles, $l = l_{12} = l_{22} = l_{32}$ and v is the manipulator height.

$$\boldsymbol{M}_P = \begin{bmatrix} m_1 & m_3 & m_e \end{bmatrix}^T$$

$$\boldsymbol{CG}_P = \begin{bmatrix} \boldsymbol{cg}_1 & \boldsymbol{cg}_3 & \boldsymbol{cg}_e \end{bmatrix}^T, \ \boldsymbol{I}_P = \begin{bmatrix} \boldsymbol{I}_{P_1} & \boldsymbol{I}_{P_3} & \boldsymbol{I}_{P_e} \end{bmatrix}$$

Generally IGM $\boldsymbol{\Theta}_P = \mathbf{F}_P^{-1}(\boldsymbol{X}_P, \boldsymbol{\xi}_P)$ for parallel manipulators can be mostly solved in closed form but the solution of DGM $\boldsymbol{X}_P = \mathbf{F}_P(\boldsymbol{\Theta}_P, \boldsymbol{\xi}_P)$ is much more difficult and it can be proven that there does not exist closed form solution and up to eight different positions of the end effector can be found for given joint coordinates. On the other hand there are some efficient numerical methods for dealing with this problem. For more details see [1].

II. Dynamic analysis for control purposes

Many control strategies for control of robotic manipulators have been developed up to now. Essentially, there are two main concepts for dealing with this problem. Firstly, complete decoupling of robot actuators is considered and a decentralized control strategy is applied on each of the robot actuator separately. The problem is reduced to motion control of single motor's shaft with variable payload. This

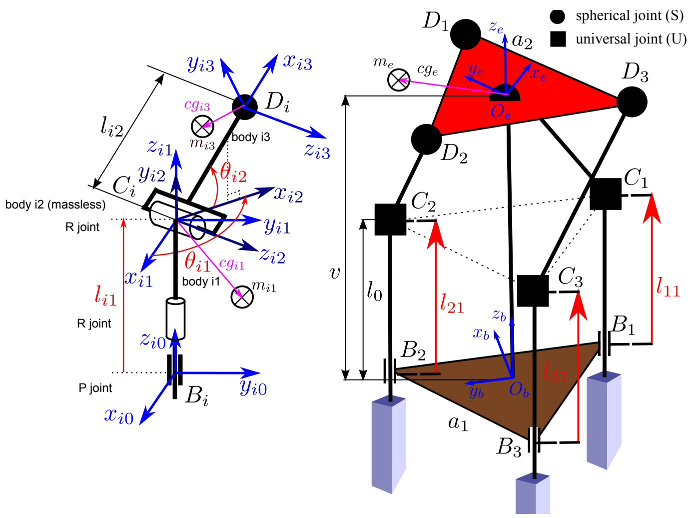

Fig. 3. Parallel part of AGEBOT and **PUS** kinematic chain

variability is caused by the change of robot configuration while performing a motion. Static and dynamic interactions among individual links of robot are neglected and it is supposed that these interactions represent uncertainties of the dynamic model of actuators. The second centralized approach is based on so called inverse dynamic model which makes possible to compute force/moment setpoints of robot actuators for given positions, velocities and accelerations of the end effector. These setpoints are supposed to be correct forces/torques for actuation of robot joints in order to follow the given end effector trajectory. Additional feedback controller is usually employed to deal with unmodelled dynamics and model uncertainties. It is clear that inverse dynamic model provides two types of forces: dynamic forces which depend on positions, velocities and accelerations of robot and static forces which are given by the gravity forces influencing the robot's links. In many cases only the static forces (gravity compensation) is sufficient for feedforward compensation for control purposes because of small required velocities and accelerations of the end effector.

A. Gravity compensation for serial manipulator

If the kinematic jacobian $\boldsymbol{J}_n = \frac{\partial \mathrm{F}_S}{\partial \boldsymbol{\Theta}_S}$ of SM is known a relationship between joint forces/moments $\boldsymbol{\tau}$ and generalized forces and moments $\boldsymbol{F}$ can be derived by virtue of a virtual work principle for the manipulator in a static equilibrium, see [2], [5], [3].

$$\boldsymbol{\tau} = \boldsymbol{J}_n^T \cdot \boldsymbol{F} \tag{5}$$

Note, that kinematic jacobians can be gained analytically without the need of symbolic derivation of $\boldsymbol{F}_S$, see [1]. For example the kinematic jacobian $\boldsymbol{J}^1_{4\,cg}$ with respect of CS F_1 can be computed from sequence of matrices $\boldsymbol{T}^1_2$, $\boldsymbol{T}^2_3$, $\boldsymbol{T}^3_{4\,cg}$ where we consider that the position of the origin of the ith link is given with respect to CS F_{i-1} as:

$$\boldsymbol{T}^{i-1}_{i\,cg} = \left[\begin{array}{c|c} \boldsymbol{R}^{i-1}_i & \boldsymbol{r}^{i-1}_{i-1,i} + \boldsymbol{R}^{i-1}_i \cdot \boldsymbol{cg}_i \\ \hline \boldsymbol{0}_{1\times 3} & 1 \end{array}\right] \tag{6}$$

Where[1] $\boldsymbol{R}^{i-1}_i = \boldsymbol{T}^{i-1}_i[1:3, 1:3]$ and $\boldsymbol{r}^{i-1}_{i-1,i} = \boldsymbol{T}^{i-1}_i[1:3, 4]$ and $\boldsymbol{cg}_i$ is the coordinates of the ith link's origin with respect to link's CS F_i.

Then it is possible to express overall static kinematic forces (**P** joint) and/or moments (**R** joints) as a sum of individual contributions of the gravity forces acting on the link's and end effector's origins in the sense of equation (5).

The first **P** joint of SM is not influenced by gravity so the static gravity compensation $\boldsymbol{\tau}_S$ for remaining joints can be expressed as:

$$\underbrace{\begin{bmatrix} M_2 \\ M_3 \\ M_4 \end{bmatrix}}_{\boldsymbol{\tau}_S} = \begin{bmatrix} \left(\boldsymbol{J}^1_{2\,cg}\right)^T & \left(\boldsymbol{J}^1_{3\,cg}\right)^T & & \\ \boldsymbol{0}_{1\times 3} & & \left(\boldsymbol{J}^1_{4\,cg}\right)^T & \left(\boldsymbol{J}^1_4\right)^T \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3} & & \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{F}^1_{g2} \\ \boldsymbol{F}^1_{g3} \\ \boldsymbol{F}^1_{g4} \\ \boldsymbol{F}^1_{ge} \end{bmatrix} \tag{7}$$

The gravity vectors with respect to CS F_1 (the vector are taken negatively because of compensation purposes):

$$\boldsymbol{F}^1_{gi} = \left[\begin{array}{ccc} 9.81 \cdot m_i & 0 & 0 \end{array}\right]^T \tag{8}$$

B. Gravity compensation of parallel manipulator

PM manipulator can be decomposed into three independent serial kinematic chains $\boldsymbol{B}_i\boldsymbol{C}_i\boldsymbol{D}_i$ (of type **PRR**) with joint coordinates $\boldsymbol{\Theta}_i = \left[\begin{array}{ccc} l_{i1} & \theta_{i1} & \theta_{i2} \end{array}\right]^T$ and the generalized coordinates are supposed to be position coordinates of the end effector connecting points $\boldsymbol{X}_i = \boldsymbol{D}_i$. The transformation of CSs of the ith kinematic chain is given by the homogeneous transformation matrices $\boldsymbol{T}^{i(k-1)}_{ik}(\boldsymbol{\Theta_i}[k])$.

A relation between the static joint forces $\boldsymbol{\tau}_P$ and moments $\boldsymbol{M}^b_e$ actuating the end effector of the PM is given in the sense of equation (5) through the kinematic jacobian $\boldsymbol{J}^b_e$ of the PM (the relation between joint velocities $\dot{\boldsymbol{\Theta}}_P$ and angular velocity of the end effector $\boldsymbol{\omega}^b_e$, it can be computed directly from kinematic jacobians $\boldsymbol{J}^{i0}_{i3}$ of individual kinematic chains), see [1]. But finding static forces for parallel manipulators in order to compensate an influence of gravity is more difficult because of dependencies among kinematic chains through the end effector. Therefore, it is not possible to compute static force compensation of the active joints of each kinematic chain separately (the gravity influence on the links) and add it to the static force which is given by the gravity actuating the end effector. Hence, the main idea for computation of static forces for PM is as follows:

1) Analogously to (7) it is possible to compute static forces and moments $\boldsymbol{\tau}_i$ of each kinematic chains. These forces and moments are caused by the mass of the chain's links.

$$\boldsymbol{\tau}_i = \begin{bmatrix} F_{i1} \\ M_{i1} \\ M_{i2} \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{J}^{i0}_{i1\,cg}\right)^T & \\ \boldsymbol{0}_{1\times 3} & \left(\boldsymbol{J}^{i0}_{i3\,cg}\right)^T \\ \boldsymbol{0}_{1\times 3} & \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{F}^{i0}_{gi1} \\ \boldsymbol{F}^{i0}_{gi3} \end{bmatrix} \tag{9}$$

where $\boldsymbol{J}^{i0}_{i1\,cg}$ and $\boldsymbol{J}^{i0}_{i3\,cg}$ are the kinematic jacobians of the ith kinematic chain with respect to CS F_{i0}.

[1] $\boldsymbol{A}[a:b, c:d]$ is sub-matrix of $\boldsymbol{A}$ which consists of $a \ldots b$ rows and $c \ldots d$ columns.

The gravity forces are supposed to be:

$$\boldsymbol{F}_{gik}^{ik} = \begin{bmatrix} 0 & 0 & 9.81 \cdot m_{ik} \end{bmatrix}^T \tag{10}$$

2) The joint forces and moments $\boldsymbol{\tau}_i$ of each kinematic chain can be recomputed to forces $\boldsymbol{F}_{\boldsymbol{D}_i}$ actuating the end effector in the connecting points $\boldsymbol{D}_i$:

$$\boldsymbol{F}_{\boldsymbol{D}_i} = \left((\boldsymbol{J}_{i3}^{i0})^T\right)^{-1} \cdot \boldsymbol{\tau}_i \tag{11}$$

3) The static forces $\boldsymbol{F}_{\boldsymbol{D}_i}$ induce the moment of the end effector as (the influence of the gravity actuating the kinematic chains):

$$\boldsymbol{M}_{e\,cg\,\text{chains}}^{b} = \sum_{i=1}^{3} \boldsymbol{R}_e^b \cdot \boldsymbol{D}_i^e \times \boldsymbol{F}_{\boldsymbol{D}_i} \tag{12}$$

where $\boldsymbol{R}_e^b$ is known rotation matrix given by the XYZ Euler angles $\boldsymbol{X}_P$ and $\boldsymbol{D}_i^e$ is known position of the end effector connecting points with respect to CS F_e.

4) The moment actuating the end effector because of an gravity is given as:

$$\boldsymbol{M}_{e\,cg}^{b} = \boldsymbol{R}_e^b \cdot \boldsymbol{cg}_e \times \boldsymbol{F}_{ge}^b \tag{13}$$

where $\boldsymbol{cg}_e$ is the origin with respect to CS F_e and the gravity force is:

$$\boldsymbol{F}_{ge}^{b} = \begin{bmatrix} 0 & 0 & 9.81 \cdot m_e \end{bmatrix}^T$$

The resulting static moment of the end effector which has to be compensated is expressed as:

$$\boldsymbol{M}_e^b = \boldsymbol{M}_{e\,cg\,\text{chains}}^{b} + \boldsymbol{M}_{e\,cg}^{b} \tag{14}$$

5) The static joint forces $\boldsymbol{\tau}_P$ corresponding to the static moment $\boldsymbol{M}_e^b$ of the end effector can be expressed in the similar way as in (5) because the inverse jacobian is known:

$$\boldsymbol{\tau}_P = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = (\boldsymbol{J}_e^b)^T \cdot \boldsymbol{M}_e^b \tag{15}$$

III. Simulation results

The simulation model of AGEBOT was created in the toolbox SimMechanics in Matlab, see [6]. The parameters of the model were obtained from CAD drawings of the real manipulator. The simulation model can be run in so-called *inverse dynamic mode* which returns required forces/moments of the joints for their given positions, velocities and accelerations. Required values of the joint coordinates and their derivatives are computed from required motion of the manipulator's end effector (given in generalized coordinates) through IGM and instantaneous IGM, see [1]. So it is possible to compare overall required forces/moments of the actuators with required *static* forces/moments of these ones, see Fig. 4, 5. This analysis plays an important role in the control design concerning feedforward compensation where the choice of suitable model of manipulator - complete inverse dynamic model or inverse static model (only a gravity compensation) - has to be taken into account in order to reduce a computational load.

A. Serial manipulator

Parameters:

$$\boldsymbol{\xi}_S = \begin{bmatrix} 0.26 & 0.67 & 0.44 & 0.84 \end{bmatrix}^T$$
$$\boldsymbol{M}_S = \begin{bmatrix} 282 & 106 & 52 & 84 & 15 \end{bmatrix}^T$$
$$\boldsymbol{CG}_S = \begin{bmatrix} -0.13 & -0.19414 & -0.13227e & -0.51987 \\ 0 & 0.000097 & 0.000010 & 0.003837 \\ 0 & 0.092685 & 0.057912 & 0.016341 \end{bmatrix}$$

$$\boldsymbol{I}_{S_2} = \begin{bmatrix} 3.80 & 0.002 & -2.67 \\ 0.002 & 11.5 & 0.004 \\ -2.67 & 0.004 & 8.06 \end{bmatrix}, \boldsymbol{I}_{S_3} = \begin{bmatrix} 0.63 & 0.001 & -0.53 \\ 0.001 & 2.11 & -0.007 \\ -0.53 & -0.007 & 1.77 \end{bmatrix},$$
$$\boldsymbol{I}_{S_4} = \begin{bmatrix} 0.99 & 0.16 & 0.39 \\ 0.16 & 10.5 & 0.0 \\ 0.39 & 0.0 & 10.4 \end{bmatrix}$$

Desired trajectory of the end effector $\boldsymbol{X}_S$ was chosen as a linear motion between the points $\boldsymbol{A}$-$\boldsymbol{F}$ with bang-bang profile of an acceleration with limited values of acceleration ($1\frac{m}{s^2}$) and velocity ($1\frac{m}{s}$)[2].

B. Parallel manipulator

Parameters:

$$\boldsymbol{\xi}_P = \begin{bmatrix} 0.1298 & 0.10108 & 0.1465 & 0.278 \end{bmatrix}^T$$
$$\boldsymbol{M}_P = \begin{bmatrix} 1.5 & 2 & 5 \end{bmatrix}^T, \boldsymbol{CG}_P = \begin{bmatrix} 0 & -0.07325 & 0 \\ 0 & 0 & 0 \\ 0.08 & 0 & 0 \end{bmatrix}$$
$$\boldsymbol{I}_{P_1} = \boldsymbol{I}_{P_3} = \boldsymbol{I}_{P_e} \begin{bmatrix} 3.80 & 0.002 & -2.67 \\ 0.002 & 11.5 & 0.00400 \\ -2.67 & 0.004 & 8.06 \end{bmatrix}$$

Required trajectory of the end effector $\boldsymbol{X}_P$ was chosen as the motion with constant γ coordinate and α, β to be changed in such a way that the z axis of CS F_e is aligned step by step with three given direction vectors. The angular acceleration is again considered to be bang bang profile with limited values of acceleration ($1\ \frac{rad}{s^2}$) and velocity ($1\ \frac{rad}{s}$).

IV. Decentralized PID position control

Common way of modelling of robotic manipulators is to consider them as a multi rigid body system with kinematical interconnections between separate links. Dynamical model can be obtained using Newton-Euler or Lagrange equations method. A manipulator with n degrees of freedom can be modeled by a set of second order equations in form:

$$\boldsymbol{M}(\boldsymbol{\Theta})\ddot{\boldsymbol{\Theta}} + \dot{\boldsymbol{\Theta}}^{\boldsymbol{T}}\boldsymbol{C}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} + \boldsymbol{G}(\boldsymbol{\Theta}) = \boldsymbol{T} - \boldsymbol{F_c}\mathbf{sgn}(\dot{\boldsymbol{\Theta}}) - \boldsymbol{F_v}\dot{\boldsymbol{\Theta}} \tag{16}$$

where $\boldsymbol{\Theta}_{n\times 1}$ denotes joints coordinates vector, $\boldsymbol{M}(\boldsymbol{\Theta})_{n\times n}$ is positive definite inertia matrix, $\boldsymbol{C}(\boldsymbol{\Theta})_{n\times n\times n}$ is tensor of centrifugal and Coriolis forces, $\boldsymbol{G}(\boldsymbol{\Theta})_{n\times 1}$ is gravity vector and $\boldsymbol{T}_{n\times 1}$ denotes torque/forces vector of joint actuators. The right side terms describe the effects of nonconservative forces caused by mechanical friction in the joints - F_v a F_c are supposed to be diagonal matrices of dimension $n \times n$ which determine an ammount of viscose and Coulomb friction.

[2]Linear motion has mixtured units due to the generalized coordinates consist of $[m]$ and $[rad]$.

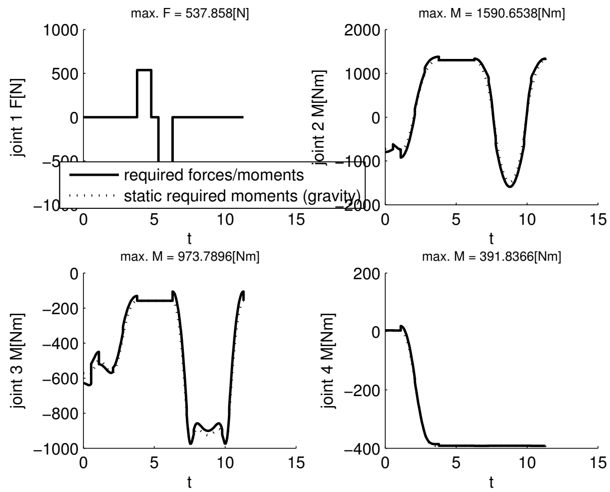

Fig. 4. Required force and moments of the joints of SM. Overall (static and dynamic) force/moments from the SimMechanics model are depicted by a solid line and static moments from algorithm mentioned above are depicted by a dash line. Note, that the first **P** joint is not influenced by a gravity.

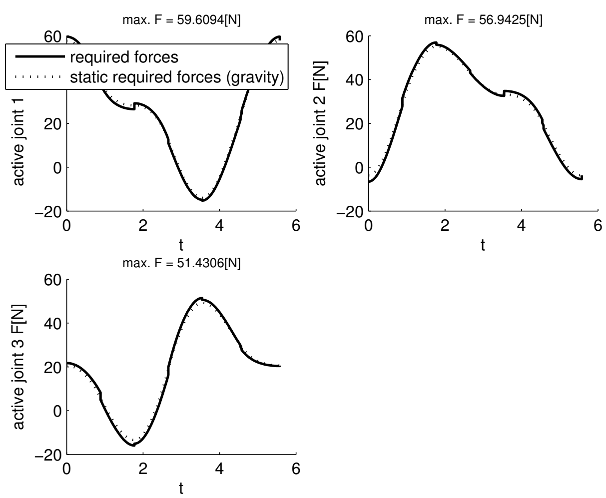

Fig. 5. Required forces of the joints of PM. Overall (static and dynamic) forces from the SimMechanics model are depicted by a solid line and static forces from algorithm mentioned above are depicted by a dash line.

Several methods have been developed for control of robotic manipulators. One of the most popular centralized control strategies called inverse dynamics control uses an idea of *global linearization* and *complete decoupling* of separate link dynamics. For the model (16), the controller can be chosen in form

$$\boldsymbol{T} = \dot{\boldsymbol{\Theta}}^{\boldsymbol{T}}\boldsymbol{C}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} + \boldsymbol{G}(\boldsymbol{\Theta}) + \boldsymbol{F_c} \boldsymbol{sgn}(\dot{\boldsymbol{\Theta}}) + \boldsymbol{F_v}\dot{\boldsymbol{\Theta}} + \boldsymbol{M}(\boldsymbol{\Theta})\boldsymbol{U} \tag{17}$$

The resulting closed loop dynamics is

$$\ddot{\boldsymbol{\Theta}} = \boldsymbol{U} \tag{18}$$

where $\boldsymbol{U}_{n\times1}$ is new input vector for completely decoupled and linear system of n double integrators. The system (18) can be stabilized for example by simple PD controller acting with the new inputs U. However, such control is seldom applicable in practice because of imperfect modelling of the robotic manipulator. Usually an additional robust controller is needed to deal with remaining dynamics and disturbances which are not precisely compensated by controller (17).

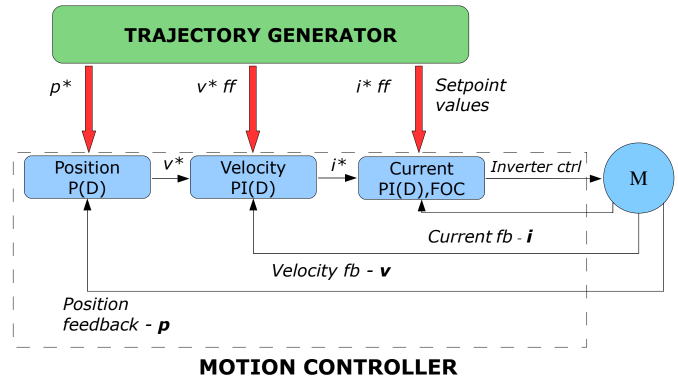

Fig. 6. Standard cascade PID control structure of an industrial servodrive

Altogether, the mentioned centralized control scheme brings high computational load for the control system which is also burdened by another tasks of motion planning and solution of kinematic transforms between the operational and joint space. Application of centralized controllers often brings a necessity of development of specialized hardware for robot control.

On the opposite side, most of the commercially available servodrives use single loop cascade PID structure for position control Fig. 6. Current loop controls the mechanical torque generated by the drive. Usually the Field oriented control scheme along with PI(D) algorithm and space vector modulation or Direct torque control method is used for driving the voltage source three-phase frequency inverter. On the next level, PI or PID velocity controller is employed. The last layer is formed by position controller which most frequently runs in proportional mode. Setpoint values are acquired from trajectory generator (interpolator) which computes desired motion for the given axis of the machine.

The above presented kinetostatic analysis of the AGEBOT manipulator leads to conclusion, that the virtual dynamic forces $\dot{\boldsymbol{\Theta}}^{\boldsymbol{T}}\boldsymbol{C}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}}$ acting on the joints along the desired trajectories are relatively small with respect to static forces $\boldsymbol{G}(\boldsymbol{\Theta})$ caused by gravity. Also the mutual dependencies of joint accelerations described by non-diagonal terms of $\boldsymbol{M}(\boldsymbol{\Theta})$ are negligible. Therefore, the robot dynamics can be simplified to obtain *static model* of the system in form:

$$\boldsymbol{M_{av}}\ddot{\boldsymbol{\Theta}} + \boldsymbol{G}(\boldsymbol{\Theta}) = \boldsymbol{T} - \boldsymbol{F_c}\mathbf{sgn}(\dot{\boldsymbol{\Theta}}) - \boldsymbol{F_v}\dot{\boldsymbol{\Theta}} \tag{19}$$

where $\boldsymbol{M_{av}}$ denotes *averaged static* inertia matrix which no longer depends on robot configuration and can be computed from expression

$$\boldsymbol{M_{av}} = \boldsymbol{I}_{n\times n} \cdot \frac{\sum_{i=1}^{n} diag(\boldsymbol{M}(\boldsymbol{\Theta_i}))}{N} \tag{20}$$

where $\boldsymbol{\Theta}_i; i = 1..N$ denotes properly chosen set of robot configurations for some typical trajectories in operational space. The simplified static model can now be used for tuning of cascade PID control structure for individual robot actuators.

We suppose that the current control loop is working properly in each of the robot drives. The motors act as torque generators and the output torque is directly proportional to desired current. The dynamics of the current loop can be neglected because of significantly shorter time constants with respect to mechanical system of the manipulator. Therefore, the transfer function from motor torque to robot link velocity Ω_i can be obtained from (19, 20):

$$P_i(s) = \frac{\Omega_i(s)}{T_i(s)} = \frac{1}{M_{av}^{i,i} s} = \frac{1}{I_i s};\ i = 1 \dots n \tag{21}$$

where $M_{av}^{i,i} = I_i$ is averaged static moment of inertia with respect to $i - th$ joint which corresponds to diagonal terms of $\boldsymbol{M_{av}}$. The gravitational and friction terms of (19) are neglected. They can be either compensated by feedforward action or they are treated as an equivalent input disturbance in form of load torque.

We suppose 2Dof PI control law in form

$$T_i(s) = K\{b\Omega_i^*(s) - \Omega_i(s) + \frac{1}{\tau s}[\Omega_i^*(s) - \Omega_i(s)]\} \tag{22}$$

with gain K, integral time constant τ, setpoint weighting factor b and desired link velocity Ω_i .

Resulting closed loop transfer function of system (21) with controller (22) is

$$F_{v_i}^{cl}(s) = \frac{\Omega_i(s)}{\Omega_i^*(s)} = \frac{\frac{Kb}{I_i}s + \frac{K}{\tau I_i}}{s^2 + \frac{K}{I_i}s + \frac{K}{\tau I_i}} \tag{23}$$

Two closed loop poles can be arbitrary assigned by the choice of controller gains. Location of the real stable closed loop zero can be changed by varying the weighting factor b. The desired location of closed loop poles of (23) can be parametrized by second order polynomial $s^2 + 2\xi\omega_n s + \omega_n^2$ with natural frequency ω_n and relative damping ξ. By solving the pole placement problem we can easily get resulting PI controller gains:

$$K = 2\xi\omega_n I_i;\ \tau = \frac{2\xi}{\omega_n} \tag{24}$$

All the neglected parts of robot dynamics in form of interactions between the links caused by inertial, centrifugal and Coriolis effects, further gravitational and friction terms can be summed into an equivalent disturbance load torque T_{li} on the input of each joint. The closed loop transfer function from this input disturbance to link velocity is

$$F_{di}^{cl}(s) = \frac{\Omega_i(s)}{T_{li}(s)} = \frac{\frac{1}{I_i}s}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{25}$$

Derivative nature of this transfer function caused by its zero shows to zero steady state error. Proper choice of desired ξ and ω_n leads to stable and well damped transient response. Position loops use proportional controllers whose gain can be derived for example by root locus method or simply by experimental tuning.

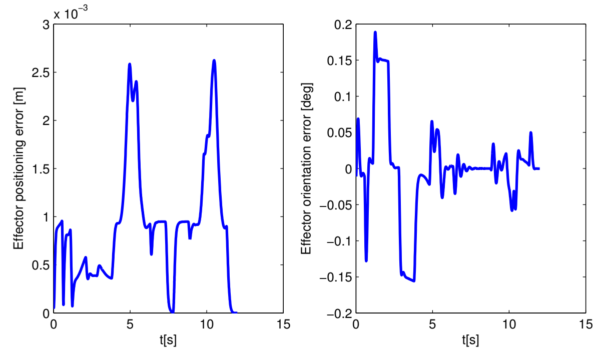

Fig. 7. End-effector positioning error using proposed control scheme

Functionality of the proposed design is demonstrated in Fig. 7. The robot tracks the trajectory from the previous example and resulting position and orientation errors are shown. The velocity controllers were tuned for values $\omega_n = 60, \xi = 0.8, b = 0.5$. Position controller gains were set to obtain highest achievable bandwidth while preserving aperiodic transient response.

V. Conclusion

The paper presents the newly developed robotic manipulator which was designed for operation in chemically aggressive environment of technology for industrial parts cleaning. Analysis of the robot showed that complete dynamical model is not necessary for control purposes and simplified static model in conjuction with simple decentralized PID control and feedforward gravity compensation can be used. General method for static model synthesis without necessity of explicit derivation of forward and inverse kinematics is also presented.

Acknowledgment

The paper was supported by grant TA01020457 from the Technology Agency of the Czech Republic.

References

[1] M. Goubej and M. Švejda, "Research and design of modular robotic manipulator for chemical aggressive environment," in *Carpathian Control Conference (ICCC), 2011 12th International*, may 2011, pp. 374 –378.
[2] B. S. L. Sciavicco, *Modelling and Control of Robot Manipulators*, 2nd ed. Springer, 2000.
[3] E. D. W. Khalil, *Modeling, Identification and Control of Robots*. Butterworth-Heinemann, 2004.
[4] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mechanics, June 1955*, vol. 22, pp. 215–221, 1955.
[5] J. P. Merlet, "Inverse kinematics," in *Parallel Robots*, ser. Solid Mechanics and Its Applications. Springer Netherlands, 2006.
[6] T. MathWorks, *SimMechanics User's Guide, www.mathworks.com*.