VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií
Ústav automatizace a měřicí techniky

Doc. Ing. František Šolc, CSc.

ROBOT MODELLING AND CONTROL

MODELOVÁNÍ A ŘÍZENÍ ROBOTŮ

SHORT VERSION OF HABILITATION THESIS

VUTIUM

BRNO 2003

KEYWORDS
Robot, robot kinematics, robot dynamics, modelling, simulation, control, adaptive control.

KLÍČOVÁ SLOVA
Robot, kinematika robotů, dynamika robotů, modelování, simulace, řízení, adaptivní řízení.

Habilitation thesis is stored at the Faculty of Electrical Engineering and Communication, office of scientific activities.

Práce je uložena na vědeckém oddělení FEKT VUT v Brně.

ISBN 80-214-2407-9
ISSN 1213-418X

CONTENTS

Author's introduction

Mr. Šolc was born 1940 in Luběnice (Czech Republic). He studied secondary modern school, specialization electrical engineering in Brno (1954-1958). He worked one year as a design engineer in aircraft industry. In 1964 he graduated at the Faculty of Aviation, Military Academy of Brno. Since 1964 he was engaged at the Military Academy of Brno as a research assistant in aircraft control and electrical equipment of aircraft. In 1966 he joined Brno University of Technology (BUT) as a research assistant in control engineering. Since 1969 he is continuing at BUT as a lecturer. In 1977 he defended PhD. thesis "Estimation of Load of an Air Traffic Control System" and received PhD in technical cybernetics.

He has delivered lectures in control engineering, system engineering, modelling and simulation, CIM and robotics. He also delivered some lectures abroad, there were lectures in control engineering in Egypt (2 years) and lectures in CIM-CAD in England (3 months). He paid short visits to universities in France, Italy, England, U.S.A., Germany etc.

He was head of research team of grant GACR "Advanced schemes of robot control" and he is head research team of current grant GACR "Research in control of smart robotic actuators". At preset he is responsible for research and pedagogical activities in robotics at the Department of Control, Faculty of El. Engg. BUT.

He is author or co-author of more than 50 papers in conferences and journals. His research interest is in control, modelling and robotics.

1 INTRODUCTION

Industrial robot is a complex electromechanical system, its effective control is still a challenging task. From systems point of view we can see four hierarchical levels of robot control.

The highest level of the control system contains elements of artificial intelligence and sometimes it is called cognitive level. At this level the control system processes information from external sensors (visual, tactile, acoustic etc.) to prepare global plan of the robot activity. Typical problem at this level can be grasping of human commands e.g. "Pick up the blue ball and put it into the yellow box", recognition that the "blue ball" and the "yellow box" is in the working space of the robot, preparation of the global activity e.g. "first open the lid of the box, then pick up the ball, avoid obstacles, put it in the box, close the lid" etc.

The next lower level is the strategic level at which the global task is divided into the elementary operations in accordance with the solution generated by the superior level. What are the elementary operations depends on the particular task of the robot. At the strategic level the most frequently planned elementary operations belong to path planning i.e. determination of trajectory to be followed by the robot gripper. At this control level the trajectory is planned with respect to the absolute co-ordinate frame which is usually fixed to the robot's base.

At the majority of today's industrial robots both the mentioned control levels are performed by a human operator.

While the motion of the gripper is planned in external co-ordinates the movement of the robot is realized via the movements of robot's particular joints. Thus to perform planned trajectory it is necessary to determine how to move individual joints of robot manipulator. This is the task of tactical level of the control system. At the tactical level the external co-ordinates of the trajectory are mapped to the robot's joint co-ordinates (internal co-ordinates). In different words at the tactical level the motion of individual joint is calculated in such a way that the final motion of the gripper follows trajectory determined by the strategic level.

Boundaries between cognitive, strategic and tactical levels are rather fuzzy and depend on the nature of the task.

The lowest level of control is executive level. Task of the executive control is to realize required motion of the individual joints to perform movement of the complete manipulator in way which was planned by the tactical control level. While the upper levels use generally information of external sensors the executive level uses generally information of inner sensors (potentiometers, resolvers, incremental encoders, force sensors etc.). Movement of joints is realized by help of actuators (electromotors, hydraulic motors, pneumatic motors). With exception of the most primitive industrial robots actuators and sensors are grouped into servosystems. From the point of view of the executive control level, a robot is a complex system with many inputs and outputs having strong interactions between individual joints. This is the main reason why the synthesis of executive control level is also complex and generally difficult task.

This paper deals only with problems of executive level and necessary problems of tactical control level which can be encountered at today's industrial robots.

In order that analysis or synthesis of executive control can be done one must be well acquainted with robot kinematics and dynamics.

2 ROBOT KINEMATICS

Robot manipulator consists of kinematic pairs. Each pair consists of two links and one joint (revolute or prismatic). Majority of industrial robots uses kinematic pair with one degree of freedom (d.o.f.) and usually six such kinematic pairs. Usually three pairs create arm of the manipulator and other tree pairs create wrist of the manipulator. A gripper is placed at the end of the manipulator wrist. Mutual position of links in kinematic pair with one d.o.f. is exactly given by one value - joint co-ordinate q. Thus position of usual robot manipulator gripper is fully characterised by the vector $\mathbf{q} = [q_1,q_2,q_3,q_4,q_5,q_6]^T$. Vectors of this type create internal or joint space. More natural for human observer is the external space or task space which is usually Cartesian space in which the position of gripper is characterised by position vector $\mathbf{p} = [x_g,y_g,z_g]^T$. Variables x_g,y_g,z_g represent usual x,y,z co-ordinates of gripper position in Cartesian system, variables, α,β,γ represent angles of roll, pitch and yaw successively see fig. 1.1. Variables $x_g,y_g,z_g,\alpha,\beta,\gamma$ represent together so called pose vector $\mathbf{s} = [x_g,y_g,z_g,\alpha,\beta,\gamma]^T$. Trajectory of. the gripper is usually planned in these co-ordinates in tactical control level.

2.1 DIRECT KINEMATICS

The first problem in kinematics is to find mapping which transforms joint co-ordinates **q** to task space co-ordinates **s**. This is done by help of homogeneous transformation. Homogeneous transform is used to describe the position and orientation of *co-ordinate frames* in space. A homogeneous transform is represented by 4x4 matrix **T** which consists of two submatrices, i.e. 3x3 rotational matrix **A** and 3x1 prismatic matrix **p** [25], [30], [31], [32], .

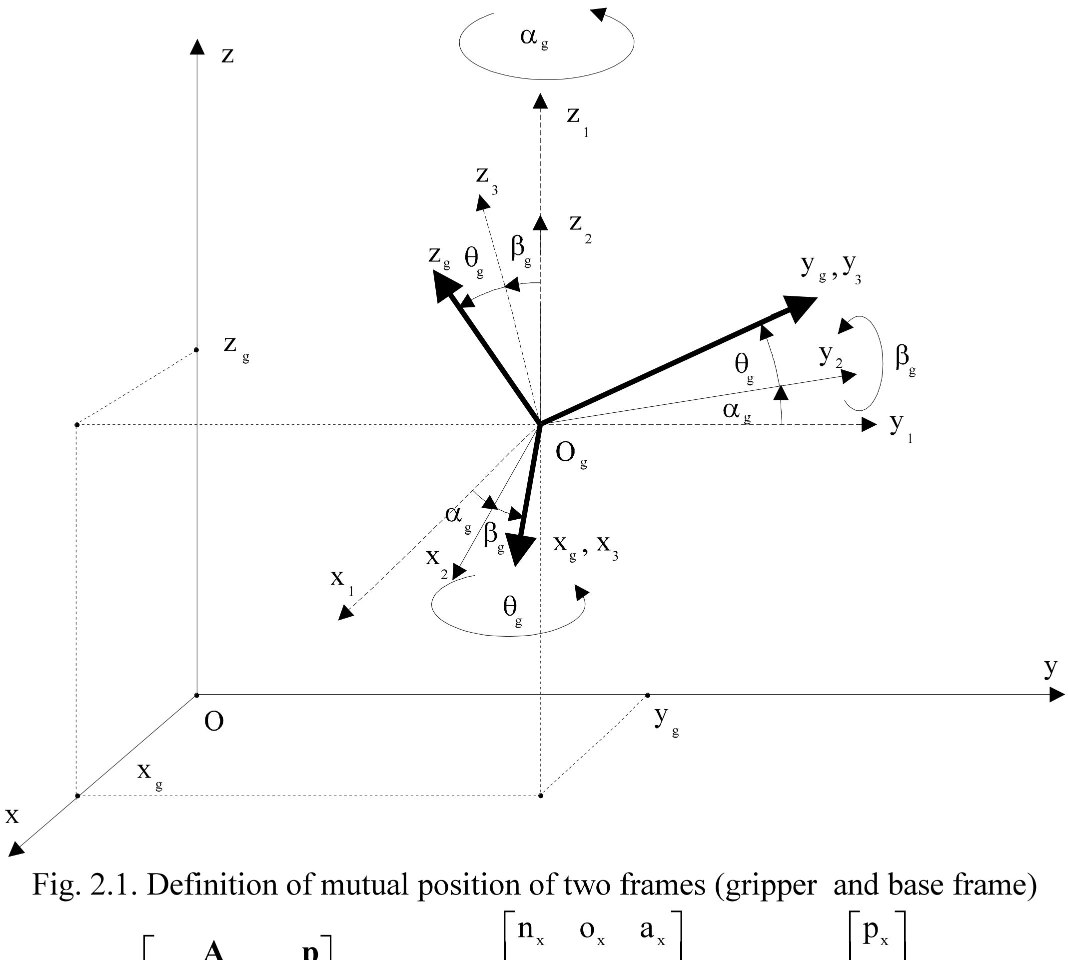

Fig. 2.1. Definition of mutual position of two frames (gripper and base frame)

$$\mathbf{T} = \begin{bmatrix} & \mathbf{A} & & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2.1}$$

From rotational matrix **A** one can find mutual rotation of frames and from prismatic matrix **p** one can find mutual translation of origins of frames. To find homogeneous transform between robot's gripper and robot's base one must

a) draw a kinematic scheme of the manipulator

b) number each link from 0 to 6 starting at the base of the manipulator as link 0 out to the last link 6
c) number the joint between link 1 and i-1 as joint i
d) allocate a frame i to each link
e) find homogeneous transforms $\mathbf{T}_{i-1,i}$ between successive links
f) find product $\mathbf{T}_{b,g} = \mathbf{T}_{0,1}\,\mathbf{T}_{1,2}\,\mathbf{T}_{2,3}\,\mathbf{T}_{3,4}\,\mathbf{T}_{4,5}\,\mathbf{T}_{5,6}$

Matrix $T_{b,g}$ represents homogeneous transform between gripper frame and base frame. Position of gripper frame origin x_g,y_g,z_g in base frame is given by values of p_x,p_y,p_z respectively. Individual angles of orientation are given [32]

$$\begin{aligned}\alpha_g &= \mathrm{atan2}(n_y,n_x)\\ \beta_g &= \mathrm{atan2}(-n_z,n_x\cos\alpha_g+n_y\sin\alpha_g)\\ \theta_g &= \mathrm{atan2}(-a_x\sin\alpha_g-a_y\cos\alpha_g,o_y\cos\alpha_g+o_x\sin\alpha_g)\end{aligned} \tag{2.2}$$

The allocation of frames to individual links should be done in a reasonable manner. Denavit Hartenberg method is recommended in [32].

2.2 INVERSE KINEMATICS

Rather more difficult is another problem, inverse kinematic problem. The task is to find inverse mapping from Cartesian space to joint space or better to say given the desired homogeneous transform between the gripper and the base, find joint co-ordinates which give this transform. Consider the situation shown in fig. 2.2. The robot is holding a peg which

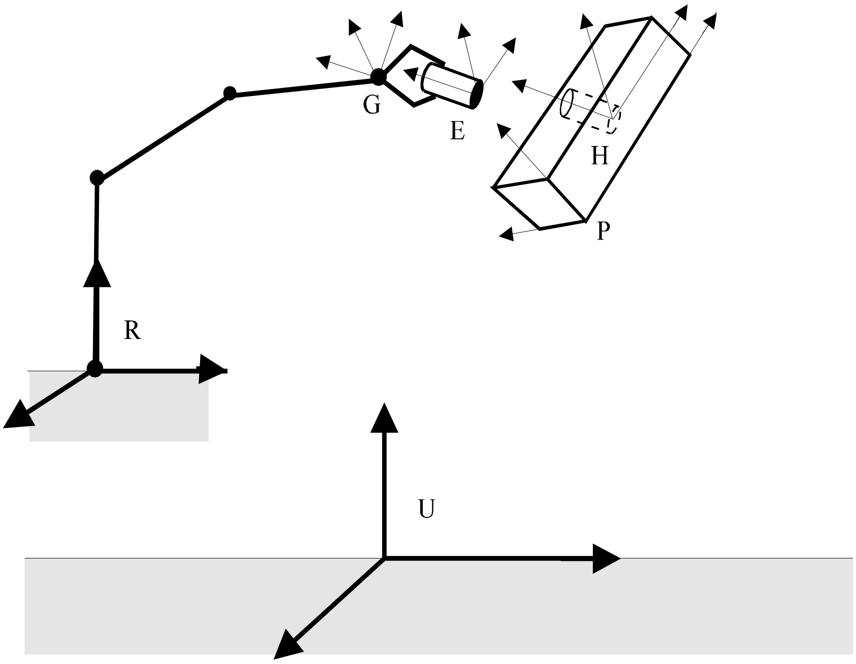

Fig. 2.2. Manipulator putting a peg into a hole

should be put in a hole in a part. This is the final desired state. The base of the robot is the origin of a frame R whose location is known to a universe frame U. That knowledge is embedded in the transformation $\mathbf{T}_{U,R}$, assumed to be constant. When in the proper location, the gripper position will be related to R by a transform $\mathbf{T}_{R,G}$. This is initially unknown. The tip of the peg is related to G by a transform $\mathbf{T}_{G,E}$, assumed constant. Thus the location of the peg tip may be related to the universe frame by

$$\mathbf{T}_{U,E} = \mathbf{T}_{U,R}\mathbf{T}_{R,G}\mathbf{T}_{G,E} \qquad (2.3)$$

Furthermore the location of the hole in the part may be related to U by

$$\mathbf{T}_{U,H} = \mathbf{T}_{U,P}\mathbf{T}_{P,H} \qquad (2.4)$$

where $\mathbf{T}_{U,P}$ represents location of part to the universe and $\mathbf{T}_{P,H}$ represents location of the hole end to the part. Thus when robot finishes its task it must be valid

$$\mathbf{T}_{U,E} = \mathbf{T}_{U,H} \qquad (2.5)$$

From these equations the desired position of gripper to base is easily calculated

$$\mathbf{T}_{R,G} = \mathbf{T}_{U,R}^{-1}\mathbf{T}_{U,P}\mathbf{T}_{P,H}\mathbf{T}_{G,E}^{-1} \qquad (2.6)$$

and tactical control must calculate joint positions accordingly.

Here we only mention that three cases are possible in relation to this problem.

a.) Unique inverse mapping can be found. Usually this is possible when number of task co-ordinates equals to number of internal co-ordinates. In majority of practical problems it requires 6 d.o.f. manipulator with gripper which axes of motion intersects in one point [32].

b.) It is not possible to find any mapping. Usually number of manipulator's d.o.f. is not sufficiently high to fulfil required task or desired position is out of the robot's task space.

c.) It is possible to find several solutions to the problem. Usually number of manipulators d.o.f. is higher than necessary for the given task

Sometimes robot control requires calculation of joint velocity and acceleration too. In such a case numerical method using Jacobian of the robot is used. Let the direct kinematics is expressed by equations

$$\mathbf{s} = \mathbf{f}(\mathbf{q}) \qquad (2.7)$$

Then we can find relation between velocities of joint and pose coordinates according to the following formula

$$\dot{\mathbf{s}} = [\frac{\partial \mathbf{f}}{\partial \mathbf{q}}]\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \qquad (2.8)$$

where $\mathbf{J}(\mathbf{q}) = [\frac{\partial \mathbf{f}}{\partial \mathbf{q}}]$ denotes the Jacobian matrix of the system f.

If the velocities of the pose vector are given, the joint velocities could be calculated from the above equation. Let **s** is *m*x*1* vector and **q** is *n*x*1* vector. Now three cases may occur :

a) *m=n*, in that case Jacobian is squared matrix and problem is solved by help of Jacobian inversion.

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\dot{\mathbf{s}} \qquad (2.9)$$

It is known that the Jacobian inversion exists except for singular positions of the robot. Various procedures for solving the singular problem have been described in the literature.

b) *m>n* , generally it is not possible to solve the problem except some special cases.

c) *m<n*, this is the case of redundant manipulators and solution of inverse problem is not unique. One of possibilities is to use so called minimal inverse solution

$$\dot{\mathbf{q}} = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\dot{\mathbf{s}} \qquad (2.10)$$

which gives solution closest to “exact” solution according to criterion of minimum quadratic error.

An inverse Jacobian can be used for numerical solution of the inverse kinematic problem with respect to position wit use of Newton method. Let we know pose s and an approximate solution of the inverse problem **q**(k) in step k. Then solution of inverse problem in step k+1 is given as

$$\mathbf{q}(k+1) = \mathbf{q}(k) - \mathbf{J}^{-1}\{\mathbf{f}(\mathbf{q}(k)) - \mathbf{s}\} \qquad (2.11)$$

The Newton method gives only one solution to the inverse kinematic problem. this solution is the closest to the initial guess **q**(k). Problems of inverse transform are discussed in detail in [32], [41]. Practical use of inverse transform is demonstrated in [37]. Speed of calculations is very important if on line computing must be used.

2.3 TRAJECTORY PLANNING

Another problem which belongs partially to tactical control is trajectory planning. Trajectory of a robot gripper can be assigned in many different ways. Generally speaking we can say that any method must produce physically feasible trajectory i.e. all accelerations and velocities of the manipulator must be attainable. The planning is usually done in Cartesian co-ordinates and when we plan the trajectory with limited acceleration and velocity of individual elements of vector s we shall receive limited joint accelerations and velocities too (if the manipulator is not in singular position where inverse kinematic gives infinitely high value of a q component). Trajectories are planned with help of control polynomials and splines very often [31],[25].

2.3.1 SPLINES FOR TRAJECTORY PLANNING

In order to approximate a Cartesian path, *m* functions of approximation are needed – one for each joint. The function of approximation for a particular joint must pass through the value calculated for that joint at each way point in Cartesian space. in addition the function should be continuous in position, velocity and acceleration in order to be physically feasible. These conditions could be met by deriving a single polynomial which passed through the way point in joint space. Such a polynomial would likely contain extrema between way points which might cause significant deviations from the desired Cartesian path.

A better solution is to define separate polynomial for each segment of the path and ensure the continuity constraints at each way point. This requires third order polynomials for the intermediate segments and fourth order polynomials for the first and the last segment, since the velocity and acceleration at the start and end points are to be zero.

Let us suppose that we have *n* way points in joint space of robot, i.e. we know *n* vectors $\mathbf{q}_k$ k=1,2,3….n. Let the dimension of the vector is 6x1 (our robot consists of 6 joints). Motion of one joint (joint *i* for example) is planned as a sequence of way points $q_{i,1}$; $q_{i,2;...}$ $q_{i,n}$ as shown in fig.2.3. (index i is omitted for simplicity).

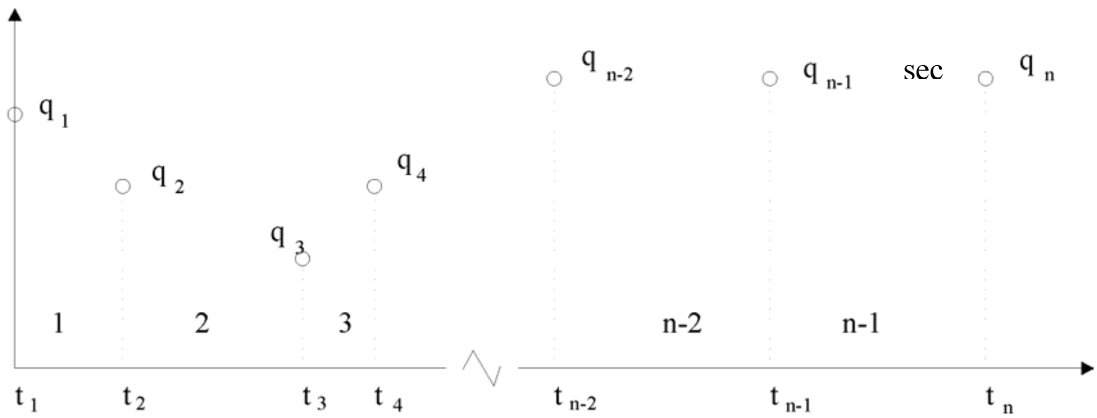

Fig. 2.3. Planned way points.

To each way point there is allocated time t_k until now tentatively, because way points are planned by teach in method usually. During teach-in planning, times at which individual way points are memorized are generally random. Thus we suppose only that $t_k<t_{k+1}$. In intervals k=1;n-1 we use fourth order polynomial for path approximation, thus

$$q_k(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3 + B_5 t^4 \qquad \text{for } k=1;n-1 \tag{2.12}$$

In other intervals we use third order polynomial, thus

$$q_k(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3 \qquad \text{for } k=2;\ldots n-2 \tag{2.13}$$

Time considered in k-th polynomial is between 0 and $\Delta_{k+1}=t_{k+1}-t_k$. Thus

$$q(t) = q_k(t - t_k) \;\text{ for }\; t \in \langle t_k ; t_{k+1} \rangle \tag{2.14}$$

Coefficients of individual polynomials are calculated I such a way that values of positions (velocities, accelerations) of polynomials q_k and q_{k+1} in t_{k+1} are equal, for k=1,2..n-2. Values of velocities and accelerations are zero for initial point of polynomial q_1 and end point of polynomial q_n.

Before we put the robot into operation according to so far planned trajectory we can accelerate or decelerate its motion with use of time scale factor. Instead using so far programmed trajectories by using functions q(t) we use q(Kt) instead. For K>1 we accelerate, for K<0 we decelerate robot motion. Value K can be tuned according velocity and/or acceleration limits valid for individual joints. Differentiating q(t) yields expression for velocity and acceleration

$$\begin{aligned} v(t) &= K \frac{dq(t)}{dt} \\ a(t) &= K^2 \frac{d^2 q(t)}{dt^2} \end{aligned} \tag{2.15}$$

Denoting V_i , A_i maximum velocity, maximum acceleration in the i-th joint respectively, the following formula for K must be valid

$$K < \min(K_v , K_a) \tag{2.16}$$

where

$$K_v = \min_i \left(\frac{V_i}{\max_k (\max_t |v_{i,k}(t)|)} \right)$$

$$K_a = \min_i \left(\frac{V_i}{\max_k (\max_t |a_{i,k}(t)|)} \right)$$

$\max_t |v_{i,k}(t)|$ represents maximum velocity of k-th polynomial of i-th joint. Similarly $\max_t |a_{i,k}(t)|$ represents maximum acceleration of k-th polynomial of i-th joint. If the maximum velocity of k-th polynomial is outside time interval of that polynomial i.e. out of <t_k , t_{k+1}>, its value need not be considered in search for maximum because of continuous acceleration requirement follows that in neighbour section the velocity i.e. k-1 or k+1 the velocity will be higher than in section k.. maximum acceleration for inner polynomials (k = 2, … n-2) - sections is estimated in their end points, maximum acceleration of the first and the last section must be searched in their definition interval.

3 ROBOT DYNAMICS

While kinematics deals with geometry and time dependent aspects of motion without considering the forces causing motion, dynamics is based on kinetics and includes the effects of forces on the motion of masses.

As it was explained above the executive control has to ensure implementation of trajectories (or, only positions) of joint co-ordinates of a manipulator e.g. trajectories given by fig.2.3. The implementation of the trajectories involves dynamic behaviour of the manipulator. In order that the executive control system may be correctly designed we must know the manipulator dynamic (differential equations describing manipulator motion).

3.1 MODEL OF MANIPULATOR

Several methods are used for dynamic model construction e.g. method based on Lagrange's equations, method based on Newton-Euler equations etc. Generally the complete model of manipulator dynamics can be expressed in the following form

$$\mathbf{H(q)\ddot{q} + h(\dot{q}, q) = P} \tag{3.1}$$

where $\mathbf{P}$ = $(P_1,P_2,P_3,P_4,P_5,P_6)^T$ represents the vector of driving forces (torques), $\mathbf{H(q)}$ is 6x6 inertia matrix which is the function of joint variables and $\mathbf{h(\dot{q}, q)}$is 6x1 vector function of joint variables and their derivatives. $\mathbf{H(q)}$ consists of moments of inertia around individual joints H_{ii} and cross-inertia terms H_{ij} which represents inertial effects of movement of j-th joint on i-th joint. $\mathbf{h(\dot{q}, q)}$ represents effects of centrifugal and Coriolis forces and effect of gravity forces. Generally the dynamic model of the manipulator represents a set of non-linear differential equations. From system point of view it is complex dynamic system with intercoupling. The following example explains briefly Lagrange's method in construction of model of manipulator from fig.3.1.

Let us suppose that mass of the manipulator is concentrated in points according to fig.3.1.

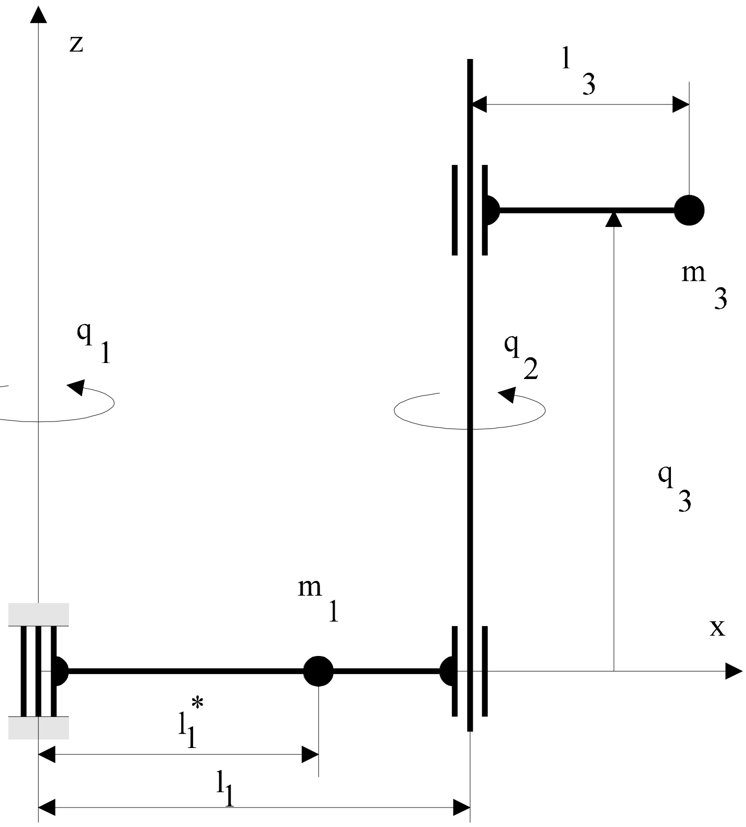

Fig.3.1. Simplified scheme of manipulator for dynamic modelling (l_1=0.6m; l_1^*=0.36m; l_3=0.2m; m_1=7kg; m_3=4kg)

Let us develop manipulator's model by help of Lagrange's equations

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} = Q_i \tag{3. 2}$$

where K is total kinetic energy of the system, q_i is i-th joint variable and Q_i is working force (torque) in i-th joint. After some calculations one can find that the kinetic energy of the system is

$$K = \{m_1(l^*_1\dot{q}_1)^2 + m_3[(l_1\dot{q}_1)^2 + l_3^2\ (\dot{q}_1+\dot{q}_2)^2 + 2l_1l_3\dot{q}_1(\dot{q}_1+\dot{q}_2)\cos q_2 + \dot{q}_3^2]\}/2 \qquad (3.3)$$

Individual values Q_i are

$$Q_1 = T_1;\ Q_2 = T_2;\ Q = F_3 - m_3 g \qquad (2.4)$$

where $T_{1,}T_2$ are torques in joint 1,2 respectively, F_3 is force in joint 3, and g is gravity acceleration.

When we do the differentiation prescribed in eq.(3.2) we can construct differential equations (3.1) where

$$\mathbf{H(q)} = \begin{bmatrix} m_1 l^{*}_1{}^2 + m_3 l_1^2 + m_3 l_3^2 + 2m_3 l_3 l_1 \cos q_2 & m_3 l_3^2 + m_3 l_3 l_1 \cos q_2 & 0 \\ m_3 l_3^2 + m_3 l_3 l_1 \cos q_2 & m_3 l_3^2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$\mathbf{h(q,\dot{q})} = \begin{bmatrix} -m_3 l_3 l_1 \dot{q}_2(\dot{q}_2 + 2\dot{q}_1)\sin q_2 \\ m_3 l_3 l_1 \dot{q}_1^2 \sin q_2 \\ m_3 g \end{bmatrix} \qquad (3.5)$$

$$\mathbf{P} = \begin{bmatrix} T_1 \\ T_2 \\ F_3 \end{bmatrix}$$

We can see that even in this very simple example the model of the system is rather complicated. If we want to represent model of the manipulator in state variable form we shall find that it is a system of the 6th order. It is strongly recommended to verify validity of the model at least by help of several examples.

3.2 MODELS OF ACTUATORS

Knowledge of dynamic model of mechanical part only is not sufficient for control synthesis. One must know also model of actuators which develop the driving torques or forces. At majority of industrial robots each joint is driven by a separate actuator. Majority of robots use D.C. permanent magnet electromotors. Generally the model of such an actuator is in form (see also fig.3.2.)

$$\begin{aligned} N_i^2 J_{Mi}\ddot{\Theta}_i + F_{vi}\dot{\Theta}_i + M_i &= C_{Mi} N_i i_i \\ L_i \dot{i}_i + R_i i_i + C_{Ei} N_i \dot{\Theta}_i &= u_i \end{aligned} \qquad (3.6)$$

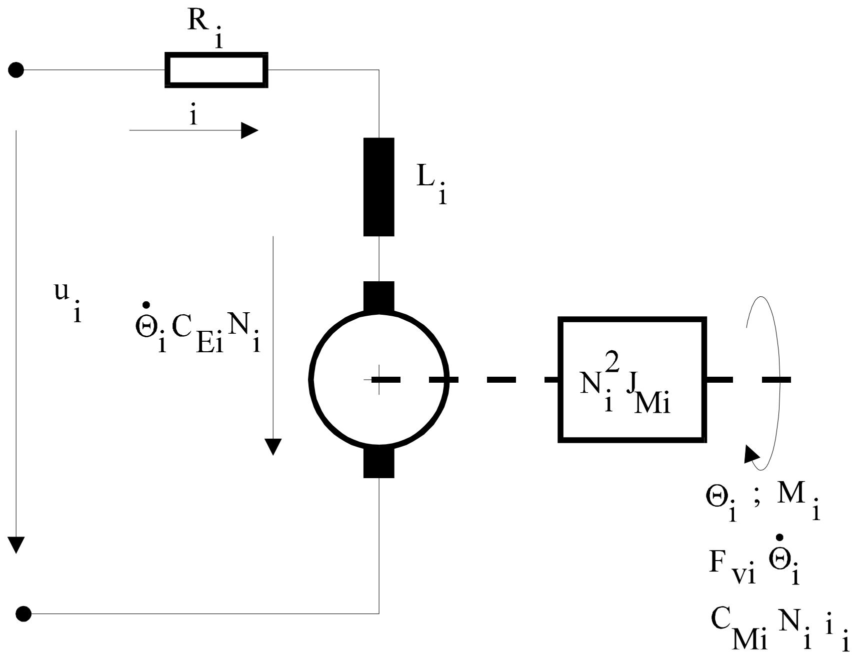

Fig.3.2. Equivalent scheme of the permanent magnet D.C. electromotor

Subscript i represents i-th joint. The first equation is equation of mechanical equilibrium at the output shaft of the gearbox which couples motor with corresponding joint. The second equation expresses electrical equilibrium in the motor's armature circuit. The meaning of used symbols is as follows: $N_i = \dot{\Theta}_{Mi} / \dot{\Theta}_i$ where Θ_i is gearbox output angle corresponding to gearbox input angle Θ_{Mi} (inverse of gear ratio), J_{Mi} is moment of inertia of the motor armature, F_{vi} is viscous damping constant reduced to gearbox output shaft, M_i is external torque at the gearbox output shaft (includes also inertial torques), C_{Mi} is torque constant of the motor, C_{Ei} is electric constant of the motor, L_i is inductance of the motor armature, R_i is resistance of the motor armature, i_i is armature current, u_i is armature control voltage. In case that rotational to prismatic motion gearbox is used one must substitute instead Θ_i, d_i which is position of gearbox output rod. No non-linear effects (Coulomb friction, backlash , non-linear gear ratio) are included in the model. These non-linear effects severely complicate the model and generally all precautions are done in order that these effects may be neglected. Anyhow one must be aware of existence of them and control system must be designed to be robust enough to suppress their effects.

Similar models can be built for other kinds of actuators.

Each actuator is generally controlled via electrical power amplifiers. Actuator together with the electrical power amplifier makes a joint drive. Generally dynamics of electrical part of the drive is negligible, but power amplifier introduces another non-linear factor - saturation which can considerably influence robot's behaviour.

Model (3.6) can be easily transformed into state variable form.

$$\dot{\mathbf{x}}_{di} = \mathbf{A}_{di}\mathbf{x}_{di} + \mathbf{b}_{di}u_i + \mathbf{f}_{di}M_i \tag{3.7}$$

Where $\mathbf{x}_{di} = [\Theta_i, \dot{\Theta}_i, i_i]^T$

$$\mathbf{A}_{di} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{F_{vi}}{N_i^2 J_{Mi}} & \frac{C_{Mi}}{N_i J_{Mi}} \\ 0 & -\frac{C_{Ei} N_i}{L_i} & -\frac{R_i}{L_i} \end{bmatrix} \tag{3.8}$$

$$\mathbf{b}_{di} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_i} \end{bmatrix} \qquad \mathbf{f}_{di} = \begin{bmatrix} 0 \\ -\frac{1}{N_i^2 J_{Mi}} \\ 0 \end{bmatrix} \tag{3.9}$$

Thus the drive is a system of the 3rd order. Generally it is possible to neglect inductance of armature of the D.C. motor, then the order of the drive model is reduced to magnitude 2.

3.3 COMPLETE DYNAMICS

Because executive control is done by help of voltages u_i we must construct integrated dynamic model input of which is vector $\mathbf{u} = [u_1,u_2,u_3,u_4,u_5,u_6]^T$ and output vector is **q** or **s**.

Let us consider the simplest case in which the movement of actuator shaft Θ_i is equal to the movement of the corresponding joint q_i i.e.

$$\Theta_i = q_i \tag{3.10}$$

In general case the relation between these two variables can be more complex.

If this simple relation is valid the following equation holds true as well

$$M_i = P_i \tag{3.11}$$

Since states of mechanical part of the manipulator are equal to some states of the actuators we can define vector of the complete system (manipulator with 6 d.o.f. and six drives) in form

$$\mathbf{x}_c = [\mathbf{x}_{c1}^T, \mathbf{x}_{c2}^T, \mathbf{x}_{c3}^T, \mathbf{x}_{c4}^T, \mathbf{x}_{c5}^T, \mathbf{x}_{c6}^T]^T \tag{3.12}$$

where individual components of $\mathbf{x}_c$ represent state vectors of drives. Thus if we consider drives to be system of the 3rd order the order of the complete system will be 18.

Individual components of the vector $\dot{\mathbf{q}}$ can be expressed according to the following equation

$$\dot{q}_i = \mathbf{k}\mathbf{x}_{ci} \tag{3.13}$$

where **k** is row vector [0,1,0] (in case of reduced model of drives i.e. model of 2nd order **k** = [0,1]).

Thus complete vector $\dot{\mathbf{q}}$ can be expressed in form

$$\dot{\mathbf{q}} = \mathbf{T}\mathbf{x}_c \tag{3.14}$$

where **T** is matrix consisting of 6 vectors k , **T**=diag(**k**)

$$\mathbf{T} = \begin{bmatrix} \mathbf{k} & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{k} & & & \\ \mathbf{0} & & . & & \mathbf{0} \\ & & & . & \mathbf{0} \\ & & & \mathbf{0} & \mathbf{k} \end{bmatrix} \tag{3.15}$$

The model of mechanical part (3.1) can be expressed using complete state vector $\mathbf{x}_c$ in the following form

$$\mathbf{H}(\mathbf{x}_c)\mathbf{T}\dot{\mathbf{x}}_c + \mathbf{h}(\mathbf{x}_c) = \mathbf{P} \tag{3.16}$$

Now we can combine model of mechanical part and model of drives together. Because of eq. (2.11) and (2.12) we can write the model of all actuators in form

$$\dot{\mathbf{x}}_c = \mathbf{A}\mathbf{x}_c + \mathbf{B}\mathbf{u} + \mathbf{F}\mathbf{P} \tag{3.17}$$

where $\mathbf{A} = \mathrm{diag}(A_{di})$; $\mathbf{B} = \mathrm{diag}(\mathbf{b}_{di})$ and $\mathbf{F} = \mathrm{diag}(\mathbf{f}_{di})$.

If we substitute $\mathbf{x}_c$ from (3.17) into (3.16) and solve (3.16) to express P we get

$$\mathbf{P} = (\mathbf{I} - \mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{F})^{-1}[\mathbf{H}(\mathbf{x}_c)\mathbf{T}(\mathbf{A}\mathbf{x}_c + \mathbf{B}\mathbf{u}) + \mathbf{h}(\mathbf{x}_c)] \tag{3.18}$$

where $\mathbf{I}$ is 6x6 unit matrix. Inverse of matrix $\mathbf{I} - \mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{F}$ always exists because control $\mathbf{u}$ can produce only physically realizable forces. Now when we substitute $\mathbf{P}$ from eq.(3.18) into eq.(3.17) we get the complete model in form

$$\dot{\mathbf{x}}_c = \mathbf{A}_c(\mathbf{x}_c) + \mathbf{B}_c\mathbf{u} \tag{3.19}$$

where

$$\mathbf{A}_c(\mathbf{x}_c)=[\mathbf{A}+\mathbf{F}(\mathbf{I}-\mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{F})^{-1}\mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{A}]\mathbf{x}_c+\mathbf{F}(\mathbf{I}-\mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{F})^{-1}\mathbf{h}(\mathbf{x}_c) \tag{3.20}$$

is 18x1 or 12x1 vector function depending on the drive model order

$$\mathbf{B}_c = \mathbf{B} + \mathbf{F}(\mathbf{I}-\mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{F})^{-1}\mathbf{H}(\mathbf{x}_c)\mathbf{T}\mathbf{B} \tag{3.21}$$

is 18x6 or 12x6 matrix depending on the drive model order.

This model of manipulator together with drives is called centralized model.

We can see that the state space model of the complete system is rather complicated. Little bit more transparent model is block scheme model from fig.3.3. Construction of this model is based on equations (3.7) and (3.1). But when we try to simulate the system expressed by the above drawn block scheme we shall find that we must solve for several algebraic loops which occur in the scheme [35]. An algebraic loop going through blocks without dynamics (blocks k, H(q), $f_{di}k_i$) is clearly seen in the scheme. To avoid solution of this problem which can result in many errors, we can use simulation of the robot with help of matrix model (3.19). Block scheme of such a model in MATLAB Simulink is shown on the figure 3.4. Figure shows modelling scheme for robot with two joints only. In Matlab function blocks ac and bc there are calculated matrices $\mathbf{A}_c$ and $\mathbf{B}_c$ which were derived in the above section of the text. Construction of these Matlab functions is very easy so that programming of the model is fast and practically without any error.

Even from these simple examples one can see urgent need for computerisation of the model development procedure. Several procedures are developed in [42],[35],[38]. One can use also MATLAB-SIMULINK which is an universal CACSD package [9].

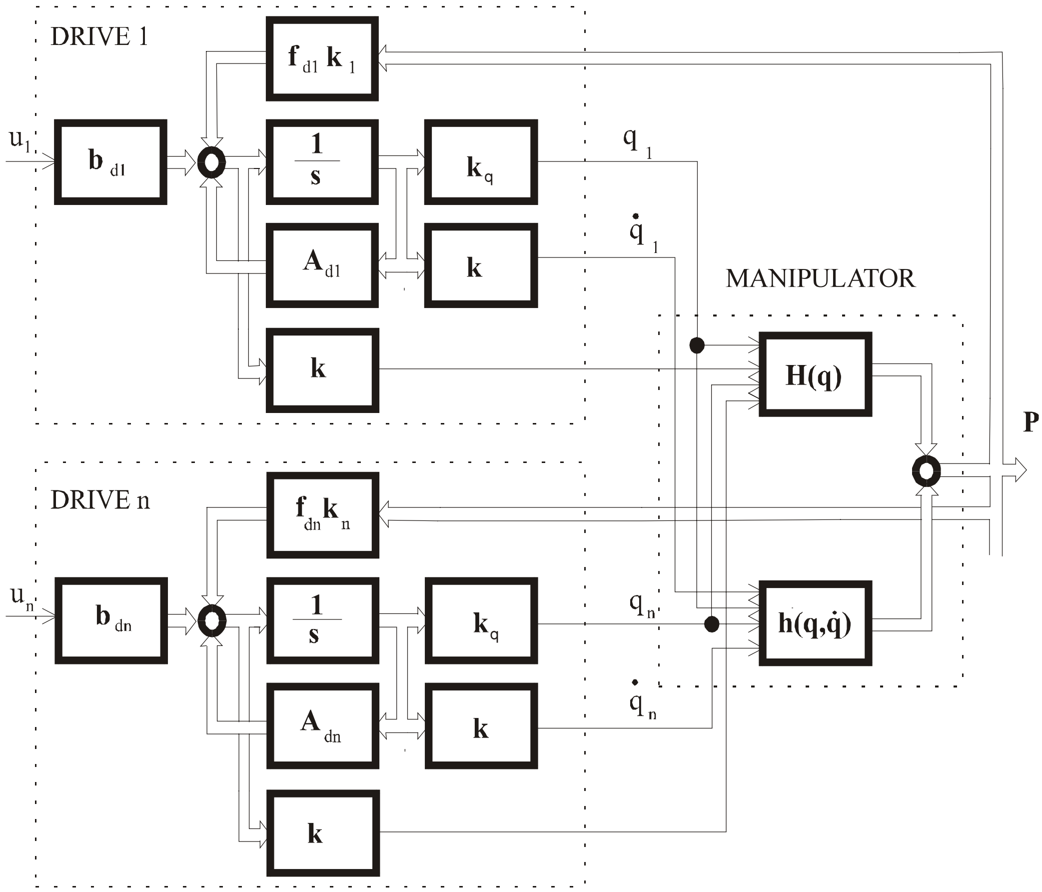

Fig.3.3. Block model of the manipulator together with drives.

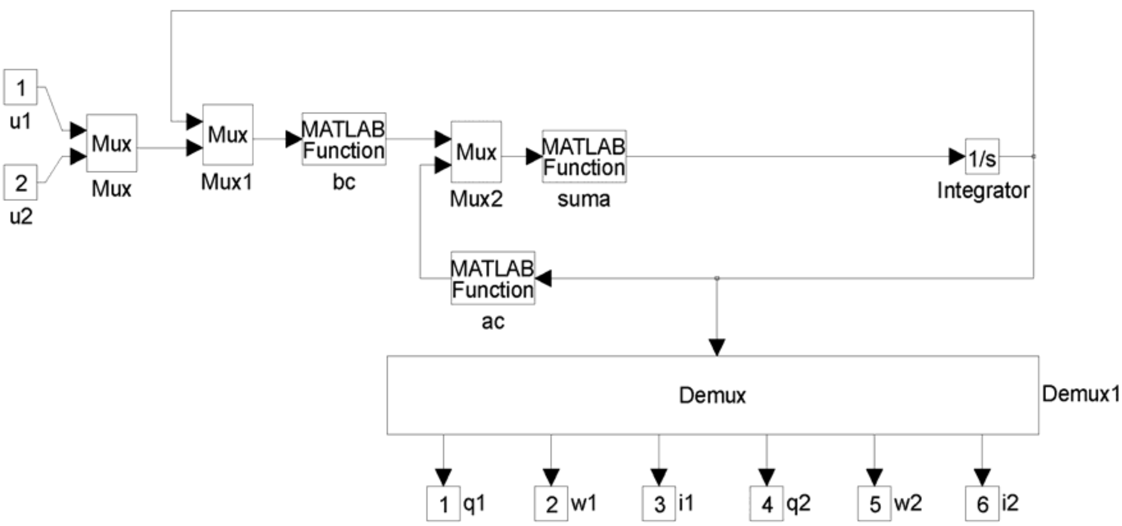

Fig.3.4. Matrix model simulation scheme

4 ROBOT CONTROL

In the following chapter we shall discuss executive robot control. As it was explained above, the task of the executive control level is to drive the robot joints into the desired positions or to drive them along a prescribed trajectories.

4.1 DECENTRALIZED ROBOT CONTROL

First of all we shall discuss the simplest control strategy which is based upon assumption that crosscoupling between individual joints is negligible, such control strategy is called dencentralized or local control. Thus control law for each joint is designed independently on movement of other joints. Simply we assume that all other joints are locked. Influence of disturbing forces stemming from movement of other joints is reduced only to gravity effects. Model of mechanical part is in this case reduced into the following form

$$H_{ii}(\mathbf{q}^*)\ddot{q}_i + h_i(\mathbf{q}^*, \dot{q}_i) = P_i \tag{4.1}$$

where $\mathbf{q}^*$ denotes fixed position of all joints (except joint i) and is constant. $h_i(\mathbf{q}^*, \dot{q}_i)$ represents gravity effects which can depend on position of all joints. The effect given by speed $\dot{q}_i$ (viscous friction in mechanical part) is generally negligible due to relatively small velocity of mechanical part. State variable diagram of the system is shown in the following figure then.

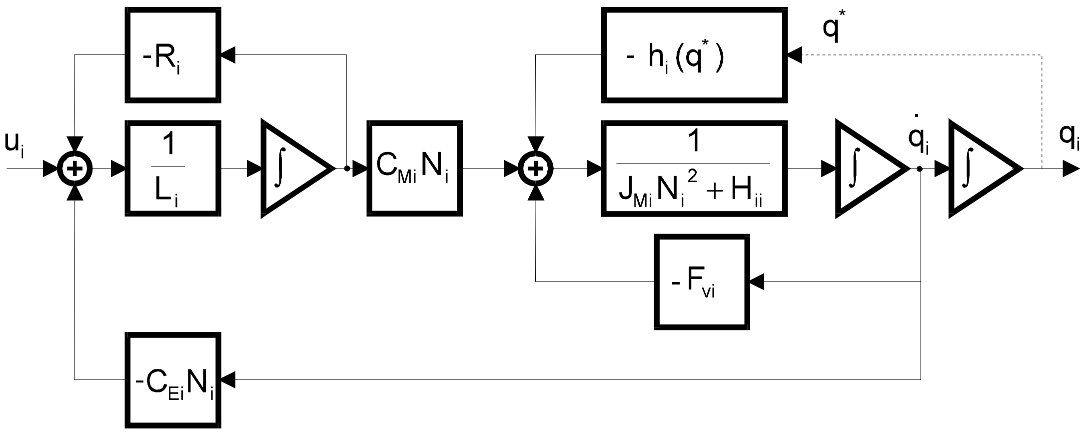

Fig.4.1. State variable diagram of the controlled system
(electrical drive plus locked mechanical part)

From state variable form we can easily derive transfer functions between individual input and output variables of the controlled system.

$$\frac{q_i(s)}{u_i(s)} = \frac{K_{di}}{[T_{ei}T_{mi}s^2 + (1 + T_{ei}F_{di})T_{mi}s + 1]s} \tag{4.2}$$

$$\frac{q_i(s)}{h_i(s)} = \frac{-(T_{ei}s + 1)K_{mi}}{\left[T_{ei}T_{mi}s^2 + (1 + T_{ei}F_{di})T_{mi}s + 1\right]s} \tag{4.3}$$

where
$T_{ei} = L_i/R_i$ is the electrical time constant of the drive

$$T_{mi} = \frac{R_i\left(J_{Mi}N_{i^2} + H_{ii}\right)}{R_iF_{vi} + C_{Mi}C_{Ei}N_i^2}$$

is the electromechanical time constant of the drive with the load H_{ii}.

$F_{di} = F_v/(J_{Mi}N_i^2 + H_{ii})$

Gains K_{di} and K_{mi} are

$$K_{di} = \frac{C_{Mi}N_i}{R_iF_{vi} + C_{Mi}C_{Ei}N_i^2} ; \qquad K_{mi} = \frac{R_iK_{di}}{C_{Mi}N_i}$$

Generally one can neglect electrical time constant T_{ei} of the drive. The influence of this time constant on dynamics of the system is usually small.

There are several control schemes which can be used for control of such decentralized system. One which is used frequently is proportional plus velocity controller with control law

$$u_i = K_p(q_d - q) - K_v\dot{q} \tag{4.4}$$

which yields the following transfer functions of the system

$$\frac{q(s)}{q_D(s)} = \frac{K_PK_d}{s(T_ms+1) + K_dK_Vs + K_dK_P} \tag{4.5}$$

$$\frac{q(s)}{h(s)} = \frac{-K_m}{s(T_ms+1) + K_dK_Vs + K_dK_P} \tag{4.6}$$

By comparison with standard second order system we shall find damping ratio, frequency of undamped oscillations and steady state error caused by external disturbance.

$$\omega_0 = \sqrt{\frac{K_PK_d}{T_m}}, \qquad \xi = \frac{1 + K_VK_d}{2\sqrt{T_mK_dK_P}} \tag{4.7}$$

$$q(\infty) = \frac{-K_m}{K_dK_P}h(\infty) \tag{4.8}$$

Generally we want to have undamped frequency of the closed loop system lower than the structural frequency ω_m of the manipulator. In case that both mentioned frequencies are close each other, resonant oscillations of the complete system will appear, which is not acceptable. Generally we want to have

$$\omega_0 \langle 0.5\,\omega_m \tag{4.9}$$

which yields

$$K_p \langle \frac{0.25\omega_m^2T_m}{K_d} \tag{4.10}$$

Knowing K_P we can calculate K_V from (4.7) to reach critical damping ratio ξ=1, for this case we get

$$K_V = \frac{2\sqrt{T_m K_P K_d} - 1}{K_d} \tag{4.11}$$

To fulfil requirement on steady state error previously derived eq. (4.19) should be used to secure $q(\infty) < e_D$.

In order that transients are without overshoots we must consider that during robot motion the moments of inertia change significantly thus during the design procedure we must choose such locked positions q^* which will secure that design requirements will be met in any other possible locked position q^+. In accordance with equations (4.7) K_p should be calculated for such position which gives the smallest value of T_m (the smallest moment of inertia). Then with fixed K_P, K_V should be calculated with the highest value of T_m (the highest moment of inertia).

Unfortunately the control scheme has not enough variable parameters to meet all requirements, but steady state error can be removed or at least suppressed by help of feedforward.

The control scheme which includes gravity error compensation is drawn in fig.4.2.

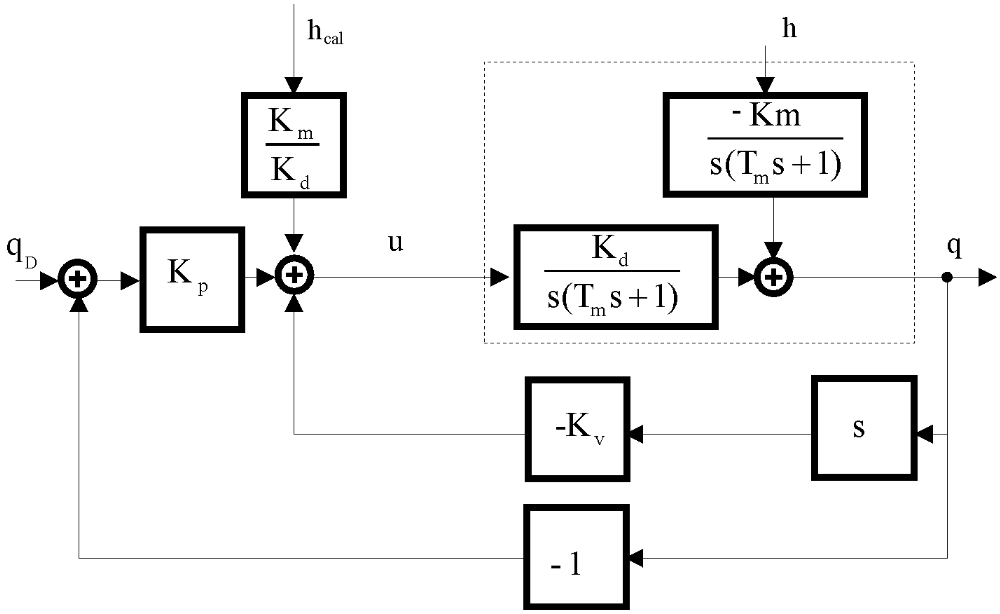

Fig.4.2. Position plus velocity control servo with direct compensation for gravity error

Simple block algebra calculations will show us that if h_{cal} (calculated disturbance) is precisely equal to real disturbance h then the effect of the disturbance on the system is zero.

Control law for such a control system is given by the following equation

$$u = K_p(q_D - q) - K_v\dot{q} + h(q^*)K_m / K_d \tag{4.12}$$

where h (q^*) is computed gravity effect on joint i.

This control scheme may work well in case that inertia crosscoupling between individual joints is negligible (generally when robot movements are slow and drives use high gear ratios) and robot is used for positioning only (PTP control).

4.1.1 DECENRALIZED CONTROL FOR TRAJECTORY TRACKING

Previous control scheme is designed to secure simple position control. Modern robots are generally designed for trajectory tracking (CP control). Desired values q_{Di} are complex functions of time in this case. It is well known that the control scheme from fig.4.2. produces constant steady state error when the desired value is linear function of time e.g. q_D= vt then

$$\lim_{t\to\infty}(q_D - q) = \frac{1+K_d K_V}{K_d K_P} v \tag{4.13}$$

When the desired value is quadratic function of time than the steady state error is infinitely high. These errors in tracking of desired trajectories are caused by delays in the servosystem. The delays can be compensated by introducing feedforward compensating signal. The simplest explanation of feedforward compensation is done by help of state equations.

Let us consider the second order model of a drive of one joint together with manipulator which has other joints locked. We shall not consider gravity effects in this case they can be compensated by already described way. State variable model of the system is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u} \tag{4.14}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & a_{22} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ b_2 \end{bmatrix} \tag{4.15}$$

$$a_{22} = -\frac{RF_v + C_E C_M N^2}{R(J_m N^2 + H_{ii})} = -\frac{1}{T_m} \qquad , b_2 = \frac{C_M N}{R(J_m N^2 + H_{ii})} = \frac{K_d}{T_m}$$

Let us try to find control u = u_D which will secure desired trajectory $\mathbf{x}_D(t)$. Certainly such control must fulfil the following equation

$$\dot{\mathbf{x}}_D = \mathbf{A}\mathbf{x}_D + \mathbf{b}\mathbf{u}_D \tag{4.16}$$

This control scheme is so called open loop control and works well only when model of system is accurate, system starts from initial conditions $\mathbf{x}(0)=\mathbf{x}_D(0)$ and there are no additional disturbances. Let us denote $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_D$; $\Delta u = u - u_D$ differences between the desired and real trajectory and between the desired and real control. Taking into account equations (4.14) and (4.16) we can write

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\,\Delta\mathbf{x} + \mathbf{b}\,\Delta\mathbf{u} \tag{4.17}$$

This state variable equation describes behaviour of the difference between desired and real state when there exists a difference between desired and real control. Introducing full state variable feedback into this system we can secure stable behaviour of $\Delta\mathbf{x}$ with steady state $\Delta\mathbf{x}$=0. Let us have

$$\Delta\mathbf{u} = \mathbf{k}^T\,\Delta\mathbf{x} \tag{4.18}$$

where $\mathbf{k}^T$=[K_P ; K_V]. Substituting (4.18) into (4.17) we get

$$\Delta\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{b}\mathbf{k}^T)\,\Delta\mathbf{x} \tag{4.19}$$

Equations (4.26) and (4.23) describe the same dynamic system. But system which described by equation (4.23) was in fact controlled by full state feedback scheme (see figs.4.1 and 4.2) thus we can calculate K_P and K_V by the same way as we did in the previous control scheme (see equations (4.10) and (4.11)). u_D can be calculated easily from equation (4.16).

$$u_D = \frac{\dot{x}_{2D} - a_{22}x_{2D}}{b_2} \tag{4.20}$$

Where x_{2D} is the second component of the vector x_D.

Thus we can write equation for control voltage of the drive

$$u = u_D + K_P(x_{1D} - x_1) + K_V(x_{2D} - x_2) \tag{4.21}$$

Substituting from (4.29) to (4.30) and taking into account that [x_1;x_2] = [q; $\dot{q}$] (the same is valid for desired values), we shall come to the following control law

$$u = \frac{\ddot{q}_D - a_{22}\dot{q}_D}{b_2} + K_P(q_D - q) + K_V(\dot{q}_D - \dot{q}) \tag{4.22}$$

Block diagram of this control scheme is drawn in fig.4.3.

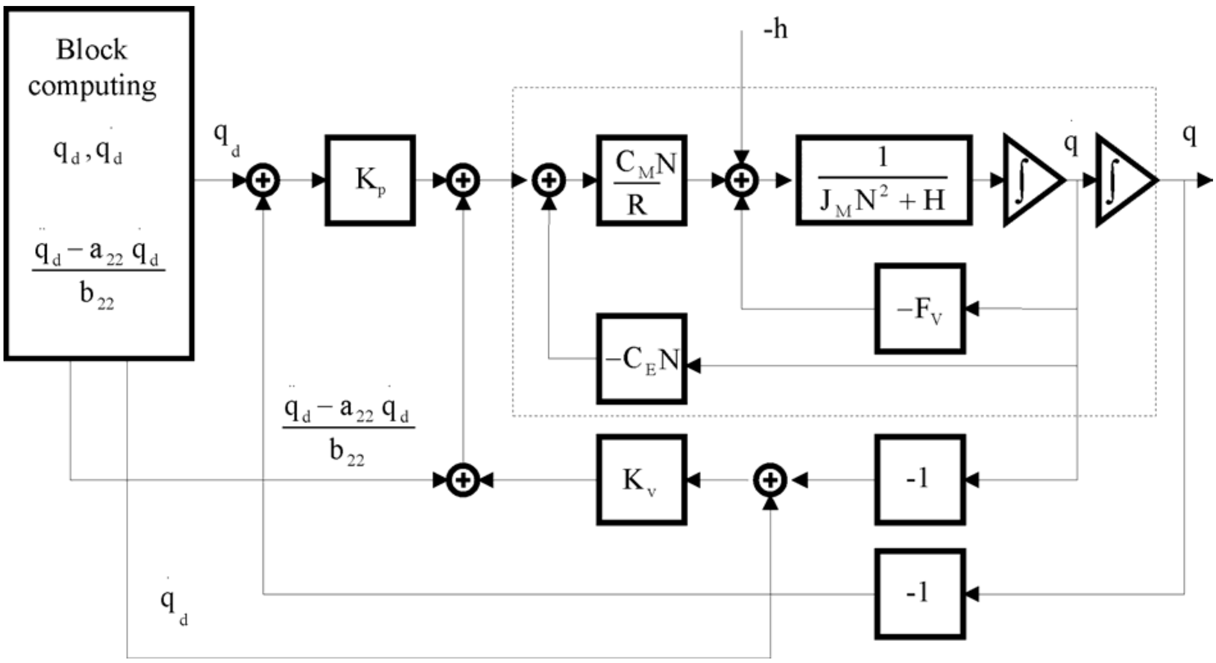

Fig.4.3. Block diagram of local servo system for trajectory tracking

Signal uD which can be calculated from equation (4.20) is called nominal control and that is why this control scheme is also called local nominal control scheme [32],[42] . The control scheme takes into account constant moment of inertia around the controlled joint and its parameters do not vary with the robot motion. While gains KP and KV should be designed for minimum and maximum moment of inertia, the local nominal control signal uD should be designed for the minimum possible value of the moment of inertia.

4.2 CONTROL OF SIMULTANEOUS MOTIONS OF ROBOT

Previous control schemes can be effectively used only when inertial crosscoupling in robot manipulator is negligible. This is in fact true only when the motion of the robot is done successively one by one joint. The condition is fulfilled approximately when the used gear ratios in drives are high. If all joints of the manipulator move simultaneously the movement of each joint generally affect the movement of other joints especially when the robot performs fast movements and used gearboxes have small gear ratios. If we use the above described decentralized control scheme for trajectory tracking we must investigate what will be the effect of crosscoupling, mainly we must investigate if it can destabilize the whole system or not. The question of stability should be investigated for the complete non-linear system, but because this is very complicated problem one must at least investigate stability of the linearized model of the complete system along desired trajectories. The linearized model of the complete controlled system can be calculated by similar manner as it was calculated the complete system model in chapter about complete dynamics. We can see that in the complete system only the manipulator is non-linear, thus we need to linearize only equation (3.1.). Let us consider nominal trajectory $\mathbf{q}_o(t)$. Along this trajectory the following equation must be fulfilled

$$\mathbf{H}(\mathbf{q}_o)\ddot{\mathbf{q}}_o + \mathbf{h}(\mathbf{q}_o, \dot{\mathbf{q}}_o) = \mathbf{P}_o \tag{4.23}$$

where $\mathbf{P}_o$ are forces which cause the nominal trajectory. Let us consider trajectory and corresponding forces which slightly differ from the nominal case i.e. $\mathbf{q} = \mathbf{q}_o + \Delta q$ and $\mathbf{P} = \mathbf{P}_o + \Delta \mathbf{P}$ thus the following equation is true

$$\mathbf{H}(\mathbf{q}_o + \Delta\mathbf{q})(\ddot{\mathbf{q}}_o + \Delta\ddot{\mathbf{q}}) + \mathbf{h}((\mathbf{q}_o + \Delta\mathbf{q}), (\dot{\mathbf{q}}_o + \Delta\dot{\mathbf{q}})) = \mathbf{P}_o + \Delta\mathbf{P} \tag{4.24}$$

Now we can express both sides of this equation in form of Taylor series along the nominal values. If we retain in the series only its linear portion we shall obtain ed model of the manipulator

$$\left[\frac{\partial \mathbf{H}}{\partial \mathbf{q}}\ddot{\mathbf{q}}_0\right]_0 \Delta\mathbf{q} + \mathbf{H}(\mathbf{q}_0)\Delta\ddot{\mathbf{q}} + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{q}}\right]_0 \Delta\mathbf{q} + \left[\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}}\right]_0 \Delta\dot{\mathbf{q}} = \Delta\mathbf{P} \tag{4.25}$$

where $\left[\frac{\partial \mathbf{H}}{\partial \mathbf{q}}\ddot{\mathbf{q}}_0\right]_0$ is matrix which i-th column is given by $\left[\frac{\partial \mathbf{H}}{\partial q_i}\right]_0 \ddot{\mathbf{q}}_0$, where $\left[\frac{\partial \mathbf{H}}{\partial q_i}\right]_0$ is matrix elements k, j of which are $\left.\frac{\partial H_{kj}}{\partial q_i}\right|_{\mathbf{q}=\mathbf{q}_0}$; $\left[\frac{\partial h}{\partial q}\right]_0$ and $\left[\frac{\partial h}{\partial \dot{q}}\right]_0$ are matrices elements k, j, of which are $\left.\frac{\partial h_k}{\partial q_j}\right|_{\mathbf{q}=\mathbf{q}_0}$; $\left.\frac{\partial h_k}{\partial \dot{q}_j}\right|_{\mathbf{q}=\mathbf{q}_0}$ successively.

Model of drives is linear, thus if we introduce for individual drive its nominal state $\mathbf{x}_{dio}$, and difference between real state and nominal state for which it is valid $\Delta \mathbf{x}_{di} = \mathbf{x}_{di} - \mathbf{x}_{dio}$ we shall receive the following state equation for difference between nominal and real state

$$\Delta\dot{\mathbf{x}}_{di} = \mathbf{A}_{di}\Delta \mathbf{x}_{di} + \mathbf{b}_{di}\Delta u_i + \mathbf{f}_{di}\Delta P_i \tag{4.26}$$

Now we can proceed by the same way as we did in calculation of the complete dynamics and we shall obtain linearized model of the complete system.

$$\Delta\dot{\mathbf{x}}_c = \mathbf{A}_{cL}\,\Delta\,\mathbf{x}_c + \mathbf{B}_{cL}\,\Delta\,\mathbf{u} \tag{4.27}$$

where

$$\mathbf{A}_{cL} = \mathbf{A} + \mathbf{F}\big(\mathbf{I} - \mathbf{H}(\mathbf{q}_0)\mathbf{T}_2\mathbf{F}\big)^{-1}\left[\left[\frac{\partial \mathbf{H}}{\partial \mathbf{q}}\ddot{\mathbf{q}}_0\right]_0 \mathbf{T}_1 + \mathbf{H}(\mathbf{q}_0)\mathbf{T}_2\mathbf{A} + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{q}}\right]_0 \mathbf{T}_1 + \left[\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}}\right]_0 \mathbf{T}_2\right]$$

$$\mathbf{B}_{cL} = \mathbf{B} + \mathbf{F}\big(\mathbf{I} - \mathbf{H}(\mathbf{q}_0)\mathbf{T}_2\mathbf{H}\big)^{-1}\mathbf{H}(\mathbf{q}_0)\mathbf{T}_2\mathbf{B}$$

Matrices $\mathbf{T}_1$=diag($\mathbf{k}_1$), $\mathbf{T}_2$=diag($\mathbf{k}_2$) are diagonal matrices of row vectors $\mathbf{k}_1$=[1,0,0] or $\mathbf{k}_1$=[1,0], $\mathbf{k}_2$=[0,1,0] or $\mathbf{k}_2$=[0,1] depending on the order of the drive model.

We can see that the linearized model (4.27) is generally linear time variant. Investigation of stability of the linear time variant systems is not an easy task as well, thus we generally investigate stability only in several typical points of the trajectory.

It is well known that stability of linearized model is the necessary condition for stability of corresponding non-linear model. Thus we can confirm only instability of the control system by this way.

Let us now investigate behaviour of robot with previously designed control schemes in trajectory tracking. Robot should follow desired trajectory $\mathbf{q}_D(t)=[q_{1D}(t);..q_{6D}(t)]^T$. When we consider second order model of drives we receive easily desired trajectory of complete state vector $\mathbf{x}_D(t)=[\,\mathbf{x}_{1D}^T(t);\ ....\ \mathbf{x}_{6D}^T(t)\,]^T = [\,q_{1D}(t);\ \dot{q}_{1D}(t);\ ..\ q_{6D}(t);\ \dot{q}_{6D}(t)\,]^T$

where $\mathbf{x}_{iD}(t) = [\,q_{iD}(t);\ \dot{q}_{iD}(t)\,]^T$.

Let us now consider control scheme with local controllers only. Local control scheme was designed under assumption that the following equations describe the real motion of the system

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i u_i + \mathbf{f}_i\overline{H}_{ii}\ddot{q}_i \tag{4.28}$$

where $\overline{H}_{ii}$ was taken as constant (the highest possible value of moment of inertia) ,then state variable feedback $u_i = \mathbf{k}_i\,\mathbf{x}_i$, where $\mathbf{k}_i = [\,-K_{pi},\ -K_{vi}\,]$ was introduced to secure desired dynamics of the system. Thus we supposed the following equations are true for each drive

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\mathbf{x}_i + \mathbf{f}_i\overline{H}_{ii}\ddot{q}_i + \mathbf{b}_i K_{Pi} q_{iD} \tag{4.29}$$

Items $\mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\mathbf{x}_i + \mathbf{f}_i\overline{H}_{ii}\ddot{q}_i$ represents the dynamics of the system. But real motion of the system is described by the following set of equations for each drive i=1,2,...6.

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\mathbf{x}_i + \mathbf{b}_i K_{Pi} q_{iD} + \mathbf{f}_i P_i(\mathbf{x}) \tag{4.30}$$

The difference between real and desired trajectory behaves according to the following equation

$$\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_{iD} = \mathbf{A}_i\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\mathbf{x}_i + \mathbf{b}_i K_{Pi} q_{iD} + \mathbf{f}_i P_i(\mathbf{x}) - \dot{\mathbf{x}}_{iD} \tag{4.31}$$

When we introduce differences $\Delta\mathbf{x}_i=\mathbf{x}_i-\mathbf{x}_{iD}$; $\Delta P_i=P_i(\mathbf{x})-P_i(\mathbf{x}_D)$, we shall obtain from (4.31) the following differential equations describing development of difference between real and desired trajectory

$$\Delta \dot{\mathbf{x}}_i = \mathbf{A}_i \Delta \mathbf{x}_i + \mathbf{b}_i \mathbf{k}_i \Delta \mathbf{x}_i + \mathbf{f}_i \Delta P_i(\mathbf{x}_D)$$
$$+ \mathbf{A}_i \mathbf{x}_{iD} + \mathbf{b}_i \mathbf{k}_i \mathbf{x}_{iD} + \mathbf{b}_i K_{Pi} q_{iD} + \mathbf{f}_i P_i(\mathbf{x}_D) - \dot{\mathbf{x}}_{iD} \quad (4.32)$$

Now we can consider equation (4.32) as a dynamic system. The first row on the right hand side of the equation represents dynamics of the difference. The second row on the right hand side of the equation can be considered as a disturbance acting upon the system. When we denote the disturbance as a 2x1 vector $\mathbf{D}_i(\mathbf{x}_D)$ and when we express $\Delta P_i(\mathbf{x}_D)$ by help of equation (4.25) than we can obtain linear state equations for difference by similar way as we did in case of equation (4.27)

$$\Delta \dot{\mathbf{x}} = \mathbf{A}_{cL} \Delta \mathbf{x} + \mathbf{B}_{cL} \mathbf{K} \Delta \mathbf{x} + \mathbf{D}_c(\mathbf{x}_D) \quad (4.33)$$

or

$$\Delta \dot{\mathbf{x}} = \mathbf{A}_T \Delta \mathbf{x} + \mathbf{D}_c(\mathbf{x}_D) \quad (4.34)$$

Where $\mathbf{A}_T = \mathbf{A}_{cL} + \mathbf{B}_{cL}\mathbf{K}$ and matrices $\mathbf{A}_{cL}$ and $\mathbf{B}_{cL}$ are the same as in equation (4.27) just instead of subscript $_o$ we use subscript $_D$. Matrix **K** is matrix of state feedback. In the main diagonal of this matrix there are row vectors $[-K_{Pi},-K_{Vi}]$ where K_{Pi} and K_{Vi} are position and velocity gains for individual drives.

Vector

$$\mathbf{D}_c(\mathbf{x}_D) = \mathbf{F}[\mathbf{I}+\mathbf{H}(\mathbf{q}_D)]^{-1}\mathbf{T}_2\mathbf{F}\mathbf{H}(\mathbf{q}_D)\mathbf{T}_2\mathbf{D} + \mathbf{D} \quad (4.35)$$

where $\mathbf{D} = \mathrm{diag}(\mathbf{D}_i(\mathbf{x}_D))$.

When we compare equations (4.32) with equation (4.29) we can see that their dynamic is very different and in fact the system (4.32) can be unstable. What will be the stability of the system during trajectory tracking we can judge from eigenvalues of system (4.33). Unless the disturbance $\mathbf{D}_c$ in (4.33) and simultaneously initial conditions are zero the system must exercise considerable differences from desired trajectory.

Now we shall investigate behaviour of local nominal control scheme. Control law for each drive is given by the following equation

$$u_i = u_{iL} + K_{Pi}(q_{iD}-q_i) + K_{Vi}(\dot{q}_{iD}-\dot{q}_i) \quad (4.36)$$

where nominal control u_{iL} is calculated from the following equation

$$\dot{\mathbf{x}}_{iD} = \mathbf{A}_i \mathbf{x}_{iD} + \mathbf{b}_i u_{iL} + \mathbf{f}_i \overline{H}_{ii} \ddot{q}_i \quad (4.37)$$

and feedback gains K_{Pi} and K_{Vi} are calculated to stabilize system

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{b}_i \mathbf{k}_i \mathbf{x}_i + \mathbf{f}_i \overline{H}_{ii} \ddot{q}_{iD} \quad (4.38)$$

where $\mathbf{k}_i = [-K_{Pi};-K_{vi}]$.

Real motion of the complete system behaves according to the following equations.

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{b}_i u_i + \mathbf{f}_i P_i(\mathbf{x}) \quad (4.39)$$

where u_i is given by equation (4.36). Now we can calculate differential equation for difference between real and desired trajectory simply by subtracting equation (4.37) from equation (4.39) and by introducing differences as in the previous case. We shall obtain the following equations

$$\Delta \dot{\mathbf{x}}_i = \mathbf{A}_i \Delta \mathbf{x}_i + \mathbf{b}_i \mathbf{k}_i \Delta \mathbf{x}_i + \mathbf{f}_i \Delta P_i(\mathbf{x}_D) + \mathbf{f}_i [\, P_i(\mathbf{x}_D) - \overline{H}_{ii} \ddot{q}_{iD}] \quad (4.40)$$

When we compare this equation with equation (4.32) we can see that we came to system with the same dynamics as in case of system (4.32) but with much simpler disturbance, namely disturbance is expressed by $\mathbf{f}_i [\, P_i(\mathbf{x}_D) - \overline{H}_{ii} \ddot{q}_{iD}]$ in our system now. When we use equation (4.25) for $\Delta P_i(\mathbf{x}_D)$ we shall obtain from (4.40) linearized equation for difference $\Delta\mathbf{x}$ which will be the same as the equation (4.34) only the disturbance effect will be different. We can see that if $P_i(\mathbf{x}_D)$ - $\overline{H}_{ii}\ddot{q}_{iD}$ = 0 i.e. crosscoupling is negligible and when we start our trajectory precisely from the desired trajectory starting point i.e. initial conditions in equation (4.40) are $\Delta x_i(0) = 0$ the difference between real and desired trajectory will be zero. But even if the disturbance is zero the dynamics of system (4.40) differs from dynamics of system (4.38) for which the feedback $\mathbf{k}_i$ was calculated. Thus the difference of real trajectory from the desired trajectory can exercise some overshoots or in some critical cases it can be unstable.

From the previous discussion one can see that it would be wise to design nominal control which would consider crosscoupling effects and to retain stabilizing feedback. Such nominal control signal $\mathbf{u}_C$ should fulfil the following equations

$$\dot{\mathbf{x}}_{iD} = \mathbf{A}_i\mathbf{x}_{iD} + \mathbf{b}_i u_{iC} + \mathbf{f}_i P_i(\mathbf{x}_D) \tag{4.41}$$

Control $\mathbf{u}_C$ is called centralized nominal control and control scheme which uses control law

$$u_i = u_{iC} + K_{Pi}(q_{iD}-q_i) + K_{Vi}(\dot{q}_{iD}-\dot{q}_i) \tag{4.42}$$

is called local control with centralized nominal control signal. Now we shall repeat the same procedure and we shall find linearized equations for difference between real and desired trajectory in case of this control law. When we subtract equations of desired motion (4.41) from equations of real motion (4.39) in which we substitute u_i from eq. (4.42) we shall obtain equation

$$\Delta\dot{\mathbf{x}}_i = \mathbf{A}_i\Delta\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\Delta\mathbf{x}_i + \mathbf{f}_i\Delta P_i(\mathbf{x}_D) \tag{4.43}$$

Again we compare this equation with the equations (4.32) and (4.40). We can see that in this case there is no disturbance in system (4.43) thus if we start with real trajectory in starting point of the desired trajectory the robot will follow the desired trajectory without any error. The dynamics of the system (4.43) is the same as the dynamics of systems (4.32) and (4.40) and we can meet similar problems with stability. If we add and simultaneously subtract from left hand part of equation (4.43) term $f_i\overline{H}_{ii}\ddot{q}_i$ we shall obtain equation

$$\Delta\dot{\mathbf{x}}_i = \mathbf{A}_i\Delta\mathbf{x}_i + \mathbf{b}_i\mathbf{k}_i\Delta\mathbf{x}_i + \mathbf{f}_i\overline{H}_{ii}\ddot{q}_i + \mathbf{f}_i[\Delta P_i(\mathbf{x}_D) - \overline{H}_{ii}\ddot{q}_i] \tag{4.44}$$

from which we can see the term $f_i[\Delta P_i(x_D)$ - $\overline{H}_{ii}\ddot{q}_i]$ which was not considered when stabilization feedback was calculated.

Finally how shall we calculate u_{ic}. For the second order model of drives we know that matrices $\mathbf{A}_i$, $\mathbf{b}_i$ and $\mathbf{f}_i$ are

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ 0 & -\dfrac{1}{J_{Mi}}\left(\dfrac{F_{Vi}}{N_i^2} + \dfrac{C_{Mi}C_{Ei}}{R_i}\right) \end{bmatrix} \tag{4.45}$$

$$\mathbf{b}_i = \begin{bmatrix} 0 \\ \dfrac{C_{Mi}}{N_i J_{Mi} R_i} \end{bmatrix} \quad \mathbf{f}_i = \begin{bmatrix} 0 \\ -\dfrac{1}{N_i^2 J_{Mi}} \end{bmatrix}$$

Thus for calculation of u_{ic} one can consider only the second row of the matrix equation (4.41). When we consider relation between vectors **x** and **q** we come to equation

$$u_{iC} = \frac{\ddot{q}_{iD} - a_{i22}\dot{q}_{iD} - f_{i2}P_i(\mathbf{x}_D)}{b_{i2}} \tag{4.46}$$

where a_{i22} is element 2,2 of matrix $\mathbf{A}_i$, and similarly b_{i2},f_{i2} are elements 2,1 of matrices $\mathbf{b}_i$ and $\mathbf{f}_i$ respectively. $P_i(\mathbf{x}_D)$ is calculated according to the following equation

$$P_i(\mathbf{x}_D) = H_i(\mathbf{q}_D)\ddot{\mathbf{q}}_D + h_i(\mathbf{q}_D;\dot{\mathbf{q}}_D) \tag{4.47}$$

where $H_i(\mathbf{q}_D)$ is the i-th row of matrix $\mathbf{H}(\mathbf{q}_D)$ and $h_i(\mathbf{q}_D;\ \dot{\mathbf{q}}_D)$ is the i-th row of the vector $\mathbf{h}(\mathbf{q}_D;\ \dot{\mathbf{q}}_D)$ from equation (3.1.).

4.3 FEEDBACK LINEARIZATION CONTROL SCHEME

The basic of feedback linearization is to construct a non-linear control law as so called inner linearization control loop which in ideal case, exactly linearizes the non-linear system after a suitable state space change of co-ordinates. Then the designer can design a second outer control loop in the new co-ordinates to satisfy the traditional linear control system design specification. Generally not all non-linear systems can be linearized, but dynamics of manipulators (3.1.) is of such a type that a relatively simple linearization method can be used. The method is described in the following text.

Let us define the control error between the desired and real trajectory in form

$$\begin{aligned} \mathbf{e}(t) &= \mathbf{q}_d(t) - \mathbf{q}(t) \\ \dot{\mathbf{e}}(t) &= \dot{\mathbf{q}}_d(t) - \dot{\mathbf{q}}(t) \\ \ddot{\mathbf{e}}(t) &= \ddot{\mathbf{q}}_d(t) - \ddot{\mathbf{q}}(t) \end{aligned} \tag{4.48}$$

Then substituting from (3.1.) for $\ddot{\mathbf{q}}$ in the last equation one comes to the following equation

$$\ddot{\mathbf{e}}(t) = \ddot{\mathbf{q}}_d(t) + \mathbf{H}^{-1}(\mathbf{q})[\mathbf{h}(\mathbf{q}(t),\dot{\mathbf{q}}(t)) - \mathbf{P}] \tag{4.49}$$

The left hand part of the equation can be taken as a control variable which governs the control error $\mathbf{e}(t)$. Thus

$$\ddot{\mathbf{e}}(t) = \mathbf{u} \tag{4.50}$$

where

$$\mathbf{u} = \ddot{\mathbf{q}}_d(t) + \mathbf{H}^{-1}(\mathbf{q})[\mathbf{h}(\mathbf{q}(t),\dot{\mathbf{q}}(t)) - \mathbf{P}] \tag{4.51}$$

Equation (4.50) can be expressed in state variable form

$$\dot{\mathbf{E}} = \mathbf{AE} + \mathbf{Bu} \tag{4.52}$$

where

$$\mathbf{E} = [\mathbf{e}(t); \dot{\mathbf{e}}(t)]^T$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_n & \mathbf{I}_n \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \qquad (4.53)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_n \\ \mathbf{I}_n \end{bmatrix}$$

Matrix E is *2n*x*1* matrix of control errors and their time derivatives. Matrices $\mathbf{0}_n$ and $\mathbf{I}_n$ are *n*x*n* zero and unit matrices respectively. Thus **A** is *2n*x*2n* matrix and **B** is *2n*x*n* matrix.

Now the designer can design classical linear control scheme in form

$$\mathbf{u} = \mathbf{KE} \qquad (4.54)$$

where **K** is *n*x*2n* matrix of control gains. Thus new dynamics of the error will be

$$\dot{\mathbf{E}} = (\mathbf{A} + \mathbf{BK})\mathbf{E} \qquad (4.55)$$

Matrix (**A**+**BK**) is to be stable, possibly with all eigenvalues real.

Combining equations (4.51),(4.54) one receives the final control law for torques an forces in form

$$\mathbf{P} = \mathbf{H}(\mathbf{q})[\ddot{\mathbf{q}}_\mathrm{d} - \mathbf{KE}] + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \qquad (4.56)$$

Time is omitted in the equation for better readability.

4.4 ADAPTIVE CONTROL SCHEMES

There are many adaptive control schemes [3]. The main idea of all adaptive control schemes is to design variable structure control to cope with uncertainty either in parameters or in payload or both together. For a given manipulator the adaptive control will find „the best“ control law from given point of view. While classical or robust control has a fixed structure which is suitable for a whole class of manipulators.

Generally all adaptive schemes can be divided into two groups.

The first group is called indirect adaptive control. The basic idea of the indirect adaptive control schemes is that the controlled plant parameters are estimated by some identification procedure, and from the estimates of these parameters, the control gains are computed at each sampling period of adaptation. This approach may be time consuming in case of robot control, because of the large number of the controlled system parameters. Thus usage of this adaptive control for real time control of speedy robots is questionable.

The second approach is the so called direct adaptive control. Direct adaptive control does not use identification of the controlled plant parameters. However the control gains are adjusted directly by an adaptation mechanism, so that the control objective is achieved. The direct adaptive control generally results in a relatively simple adaptive law, as for complexity of computing algorithms. Thus this approach is more convenient for control of robots then the first one. In the design of the direct adaptive control two approaches can be used, namely the Lyapunov direct method and hyperstability theory.

In the following chapter, adaptive control scheme based on the Lyapunov direct method will be described. Full non-linear model of the manipulator will be used in the analysis and design of the control scheme. The choice of the control law is motivated by the classical and computed torque control philosophy.

Dynamic model of manipulator will be described similarly as in the previous chapters

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_\mathrm{v}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{P} \qquad (4.57)$$

4.4.1 REFERENCE MODEL ASSUMPTION - GAIN SCHEDULING

With the reference model adaptive control approach, a reference model described by the following set of differential equations is assumed

$$\ddot{\mathbf{q}}_d(t) + \mathbf{A}_{m1}\dot{\mathbf{q}}_d(t) + \mathbf{A}_{m0}\mathbf{q}_d(t) = \mathbf{B}_{m1}\mathbf{r}(t) \tag{4.58}$$

where $\mathbf{A}_{m0}$, $\mathbf{A}_{m1}$, $\mathbf{B}_{m1}$ are nxn matrices. The matrices are chosen such that the reference model is stable (possibly with all eigenvalues real) . The reference input $\mathbf{r}$(t) is nx1 vector of continuous functions, $\mathbf{q}_d$ are vectors of desired position, velocity and acceleration with dimension nx1.

The reference model can be written also in state space form

$$\dot{\mathbf{X}}_m(t) = \mathbf{A}_M\mathbf{X}_m(t) + \mathbf{B}_M\mathbf{r}(t) \tag{4.59}$$

where $\mathbf{X}_m$(t) is the $2n$x1 state vector defined as

$$\mathbf{X}_m(t) = (\mathbf{q}_d(t)^T, \dot{\mathbf{q}}_d(t))^T \tag{4.60}$$

$\mathbf{A}_M$ is the $2n$x$2n$ matrix in form

$$\mathbf{A}_M = \begin{bmatrix} \mathbf{O}_n & \mathbf{I}_n \\ -\mathbf{A}_{m0} & -\mathbf{A}_{m1} \end{bmatrix} \tag{4.61}$$

and $\mathbf{B}_M$ is the $2n$xn matrix in form

$$\mathbf{B}_M = \begin{bmatrix} \mathbf{O}_n \\ \mathbf{B}_{m1} \end{bmatrix} \tag{4.62}$$

where $\mathbf{O}_n$ and $\mathbf{I}_n$ are zero and unit matrices of size nxn respectively.

Let us define the position tracking error vector

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \tag{4.63}$$

Similarly the velocity and acceleration errors nx1 vectors can be defined.

To drive the errors to zero, the applied torque $\mathbf{P}$(t) is generated as

$$\mathbf{P}(t) = \mathbf{K}_p(t)\mathbf{q}(t) + \mathbf{K}_v(t)\dot{\mathbf{q}}(t) + \mathbf{K}_r(t)\mathbf{r}(t) + \mathbf{P}_c(t) \tag{4.64}$$

where $\mathbf{K}_p$(t) and $\mathbf{K}_v$(t) are the nxn time varying position and velocity feedback matrices, $\mathbf{K}_r$(t) is the nxn time varying feedforward matrix, and $\mathbf{P}_c$(t) is the compensating torque vector. The time feedback, feedforward and compensating vectors will be adjusted according to an adaptive algorithm.

Applying the controller (4.64) to the manipulator dynamic model the closed loop system equation will be

$$\ddot{\mathbf{q}} - \mathbf{H}^{-1}(\mathbf{K}_v - \mathbf{C}_v)\dot{\mathbf{q}} - \mathbf{H}^{-1}\mathbf{K}_p\mathbf{q} = \mathbf{H}^{-1}\mathbf{K}_r\mathbf{r} + \mathbf{H}^{-1}(\mathbf{P}_c - \mathbf{h}) \tag{4.65}$$

From the reference model and the closed loop equations the position error satisfies the following differential equation

$$\begin{aligned} &\ddot{\mathbf{e}} + \mathbf{A}_{m1}\dot{\mathbf{e}} + \mathbf{A}_{m0}\mathbf{e} = \\ &-\mathbf{B}[(\mathbf{K}_p + \mathbf{H}\mathbf{A}_{m0})\mathbf{q} + (\mathbf{K}_v + \mathbf{H}\mathbf{A}_{m1} - \mathbf{C}_v)\dot{\mathbf{q}} + (\mathbf{K}_r - \mathbf{H}\mathbf{B}_{m1})\mathbf{r} + (\mathbf{P}_c - \mathbf{h})] \end{aligned} \tag{4.66}$$

where $\mathbf{B}=\mathbf{H}^{-1}(\mathbf{q})$.

It can be seen that position error will become zero if the right side of the error equation will be zero, that is if

$$\mathbf{K}_p = -\mathbf{H}(\mathbf{q})\mathbf{A}_{m0}$$

$$\mathbf{K}_v = -\mathbf{H}(\mathbf{q})\mathbf{A}_{m1} + \mathbf{C}_v(\mathbf{q},\dot{\mathbf{q}}) \qquad (4.67)$$

$$\mathbf{K}_r = \mathbf{H}(\mathbf{q})\mathbf{B}_{m1}$$

$$\mathbf{P}_c = \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})$$

The structure of this type of controller is illustrated in fig.3.18.

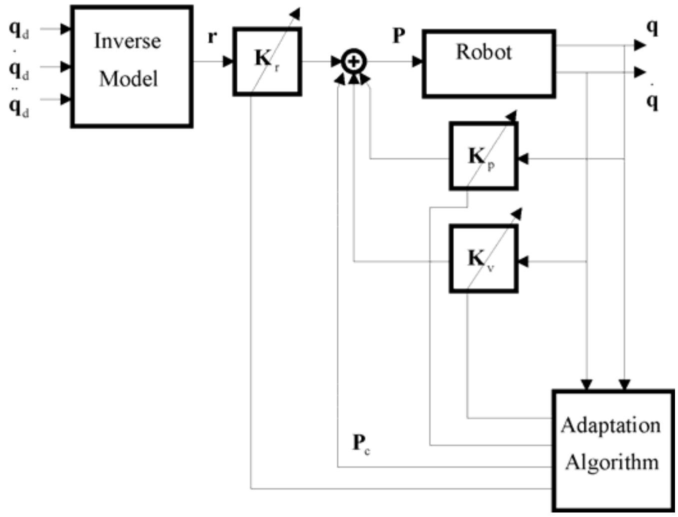

Fig.4.4. Adaptive control with inverse model reference feedforward

Obviously adjusting the control gains in the described way requires a knowledge of the manipulator parameters in order to compute $\mathbf{H}(\mathbf{q})$, $\mathbf{C}_v(\mathbf{q},\dot{\mathbf{q}})$ and $\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})$.

4.4.2 MODEL REFERENCE ADAPTIVE CONTROL - MRAC

Let $\mathbf{K}_{p0}$, $\mathbf{K}_{v0}$, $\mathbf{K}_{r0}$, and $\mathbf{P}_{c0}$ are ideal values of control gains and compensation torques or forces. The differences between real values and ideal values of control gains can be written as

$$\mathbf{K}^*_p = \mathbf{K}_p - \mathbf{K}_{p0},\ \mathbf{K}^*_v = \mathbf{K}_v - \mathbf{K}_{v0},\ \mathbf{K}^*_r = \mathbf{K}_r - \mathbf{K}_{r0}. \qquad (4.68)$$

The difference between real and ideal values of compensation torques can be written in form

$$\mathbf{K}^*_c\mathbf{i}_n = \mathbf{P}_c - \mathbf{P}_{c0} \qquad (4.69)$$

where $\mathbf{K}^*_c$ is *n*x*n* matrix and $\mathbf{i}_n$ is *n*x*1* unit vector $\mathbf{i}_n=[1,0,....0]^T$. Then the error equation can be written as

$$\ddot{\mathbf{e}} + \mathbf{A}_{m1}\dot{\mathbf{e}} + \mathbf{A}_{m0}\mathbf{e} = -\mathbf{B}[\mathbf{K}^*_p\mathbf{q} + \mathbf{K}^*_v\dot{\mathbf{q}} + \mathbf{K}^*_r\mathbf{r} + \mathbf{K}^*_c\mathbf{i}_n] \qquad (4.70)$$

The error equation can be written in state space form as follows

$$\dot{\mathbf{E}}(t) = \mathbf{A}_M\mathbf{E}(t) + \mathbf{B}_d(t)\mathbf{K}(t)\mathbf{w}(t) \qquad (4.71)$$

where $\mathbf{E}(t)$ is the *2n*x*1* state vector defined as

$$\mathbf{E}(t) = [\mathbf{e}^T(t), \dot{\mathbf{e}}^T(t)]^T \qquad (4.72)$$

$\mathbf{A}_M$ is *2n*x*2n* matrix defined above and $\mathbf{B}_d(t)$ is *2n*x*n* matrix defined as

$$\mathbf{B}_d(t) = \begin{bmatrix} \mathbf{0}_n \\ -\mathbf{B}(t) \end{bmatrix} \tag{4.73}$$

K(t) is the *n*x*4n* parameter matrix defined in form

$$\mathbf{K}(t)=[\mathbf{K}_0, \mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3] \tag{4.74}$$

where $\mathbf{K}_0 = \mathbf{K}^*_p$, $\mathbf{K}_1 = \mathbf{K}^*_v$, $\mathbf{K}_2 = \mathbf{K}^*_r$, $\mathbf{K}_3 = \mathbf{K}^*_c$

w(t) is *4n*x*1* signal vector defined as

$$\mathbf{w}(t) = [\mathbf{w}_0^T;\mathbf{w}_1^T;\mathbf{w}_2^T;\mathbf{w}_3^T]^T \tag{4.75}$$

where

$\mathbf{w}_0 = \mathbf{q}(t)$, $\mathbf{w}_1 = \dot{\mathbf{q}}(t)$, $\mathbf{w}_2 = \mathbf{r}(t)$, $\mathbf{w}_3 = \mathbf{i}_n$

Let a positive definite Lyapunov function V be chosen in form

$$V = \mathbf{E}^T\mathbf{P}\mathbf{E} + \sum_{i=0}^{3} Tr(\mathbf{M}_i^T\mathbf{B}\mathbf{F}_i^{-1}\mathbf{M}_i) \tag{4.76}$$

where $\mathbf{M}_i$ is defined as

$$\mathbf{M}_i = \mathbf{K}_i - \mathbf{L}_i \tag{4.77}$$

P is a *2n*x*2n* symmetric positive definite matrix, $\mathbf{F}_i$, i=0,1,..3 are *n*x*n* weighting matrices chosen such that $\mathbf{BF}^{-1}$ is symmetric and positive definite, $\mathbf{L}_i$, i=0,1,..3 are *n*x*n* matrices which will be chosen later. The argument t is dropped for convenience.

Since the rate of change of the robot dynamic in each adaptation sampling interval is much slower than that of the controller gains in the adaptive controller, it is reasonable to assume that the matrices from the manipulator equation are constant i. e. **B** = **const.**, $\mathbf{C}_v$ = **const.**, **h** = **const**. For this case the time derivative of the Lyapunov function along any trajectory yields

$$\dot{V} = \mathbf{E}^T(\mathbf{A}_M^T\mathbf{P} + \mathbf{P}\mathbf{A}_M)\mathbf{E} + 2\sum_{i=0}^{3} Tr([\dot{\mathbf{M}}_i^T\mathbf{B}\mathbf{F}_i^{-1} - \mathbf{w}_i\mathbf{z}^T\mathbf{B}]\mathbf{K}_i) - 2\sum_{i=0}^{3} Tr(\dot{\mathbf{M}}_i^T\mathbf{B}\mathbf{F}_i^{-1}\mathbf{L}_i) \tag{4.78}$$

where **z**(t) is the *n*x*1* weighted error vector defined as

$$\mathbf{z} = [\ \mathbf{O}_n\ \ \mathbf{I}_n\ ]\mathbf{P}\mathbf{E} \tag{4.79}$$

Where $\mathbf{O}_n$, $\mathbf{I}_n$ are *n*x*n* zero and unit matrices respectively.

From the Lyapunov stability theory is known that the necessary condition which guarantees stability of the dynamic system (in our case error **E**) is that the time derivative of the Lyapunov function is negative definite.

Thus let the matrix **P** be the solution of the Lyapunov equation

$$\mathbf{A}_M^T\mathbf{P} + \mathbf{P}\mathbf{A}_M = -\mathbf{Q} \tag{4.80}$$

where **Q** is a *2n*x*2n* positive definite matrix. Now let us chose $\mathbf{L}_i$; $\mathbf{M}_i$, i = 0,1,..3 such that

$$\mathbf{L}_i = \mathbf{G}_i\mathbf{z}\mathbf{w}_i^T \tag{4.81}$$

where $\mathbf{G}_i$, i = 0,1,...3 are weighting matrices chosen such that any $\mathbf{BG}_i$ is at least positive semidefinite matrix, and

$$\dot{\mathbf{M}}_i = \mathbf{F}_i\mathbf{z}\mathbf{w}_i^T \tag{4.82}$$

Then the time derivative of the Lyapunov function (4.76) yields

$$\dot{V} = -\mathbf{E}^T\mathbf{Q}\mathbf{E} - 2\sum_{i=0}^{3} Tr(\mathbf{w}_i\mathbf{z}^T\mathbf{B}\mathbf{G}_i\mathbf{z}\mathbf{w}_i^T) \tag{4.83}$$

which is negative definite. One can observe that this result is true even if $\mathbf{G}_i = \mathbf{0}$, but if $\mathbf{B}\mathbf{G}_i$ is positive definite then the time derivative of the Lyapunov function becomes more negative definite. Now using (4.77), (4.81), and (4.75) the following adaptive algorithm is obtained

$$\begin{aligned} \mathbf{K}_p(t) &= \mathbf{M}_0(t) + \mathbf{G}_0(t)\mathbf{z}\mathbf{q}^T(t) \\ \mathbf{K}_v(t) &= \mathbf{M}_1(t) + \mathbf{G}_1(t)\mathbf{z}\dot{\mathbf{q}}^T(t) \\ \mathbf{K}_r(t) &= \mathbf{M}_2(t) + \mathbf{G}_2(t)\mathbf{z}\mathbf{r}^T(t) \\ \mathbf{P}(t) &= \mathbf{m}_3(t) + \mathbf{G}_3\mathbf{z} \end{aligned} \tag{4.84}$$

where

$$\begin{aligned} \dot{\mathbf{M}}_0(t) &= \mathbf{F}_0\mathbf{z}\mathbf{q}^T(t) \\ \dot{\mathbf{M}}_1(t) &= \mathbf{F}_1\mathbf{z}\dot{\mathbf{q}}^T(t) \\ \dot{\mathbf{M}}_0(t) &= \mathbf{F}_0\mathbf{z}\mathbf{r}^T(t) \\ \dot{\mathbf{m}}_3(t) &= \mathbf{F}_3\mathbf{z} \end{aligned} \tag{4.85}$$

The structure of this type of controller is illustrated in fig.4.5.

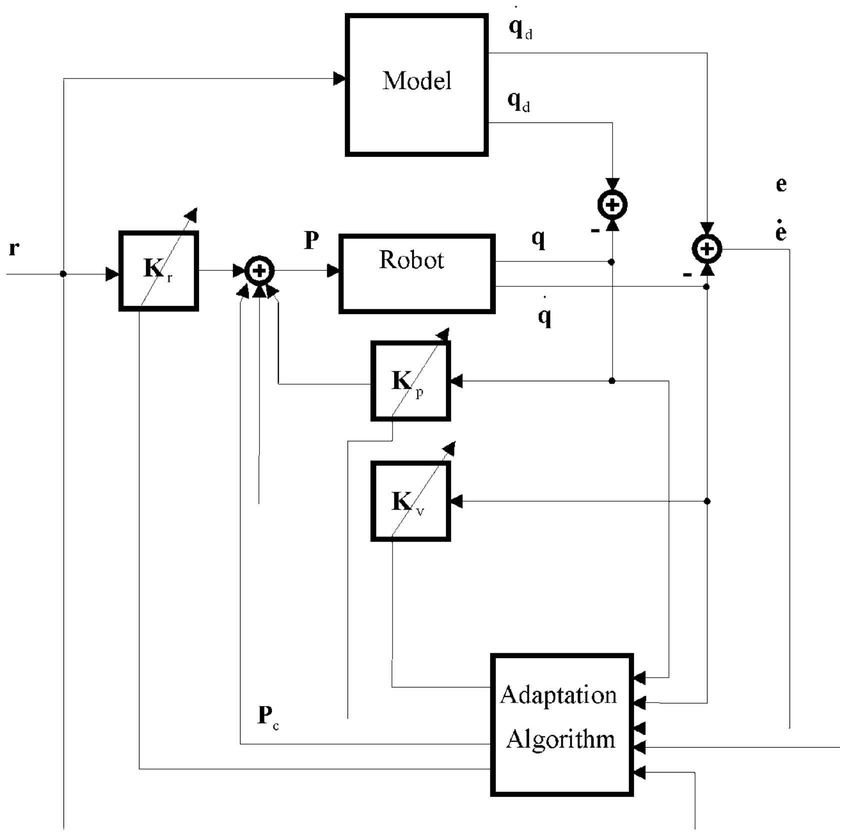

Fig.4.5. Structure of the model reference adaptive control MRAC.

5 CONCLUSIONS

The work describes and explains methods of modelling and control of industrial robots i.e. rigid mechanical manipulators, which are mainly controlled by help of electrical drives.

Basic model of mechanical part is given in form of set of differential equations that is developed by help of Lagrange equations. These equations describe time evolution of mechanical systems subjected to holonomic constraints, when the constraint forces satisfy the principle of virtual work. Simulation of the resulting system of second order differential equations brings a problem of algebraic loop. Usually these loops are to be solved analytically or numerically before simulation of the whole system, which is a source of many mistakes. Matrix model described in the work solves the problem and enables relatively simple simulation of the complex system. The matrix model enables also elegant inclusion of model or equations of drives into the complete model of the robot. Such a complete matrix model may be used for control system analysis and/or design then.

In the work there are described some of the classical and modern control schemes of robot control. The intent was not to give an exhaustive treatment of all methods of robot control as practically any control techniques are mentioned in the research literature on the subject. Many of the control methods are taken from area of process control and applied for robot control, but they have minor practical usage in area of robotics either because of unacceptable chattering of control variable or because of long computing times of complicated control algorithm. The intent of the work is to investigate only several methods that are of practical use and to perform analysis of their behaviour with help of developed matrix modelling technique. All the described control schemes were tested by simulation with help of MATLAB-Simulink and proved to be of practical use.

6 REFERENCES

[1] Arimoto, S.; Naniwa, T.; Ozaki, F.; Whitcomb, L.L.;
Adaptive model-based hybrid control of geometrically constrained robot arms
Robotics and Automation, IEEE Transactions on , Volume: 13 Issue: 1 , Feb 1997
[2] Arteaga, M.A.; Yu Tang;
Adaptive control of robots with an improved transient performance
Automatic Control, IEEE Transactions on , Volume: 47 Issue: 7 , Jul 2002
[3] Astrom, K.J.; Wittenmark, B.
Adaptive control
Addison-Wesley Pub Co, 1995
[4] Berghuis, H.; Nijmeijer, H.; Ortega, R.;
A robust adaptive robot controller
Robotics and Automation, IEEE Transactions on , Volume: 9 Issue: 6 , Dec 1993
[5] Bito, T.; Kanzaki, K.; Kawasaki, H.;
An efficient algorithm for the model-based adaptive control of robotic manipulators
Robotics and Automation, IEEE Transactions on , Volume: 12 Issue: 3 , Jun 1996
[6] Bin Yao; Li Xu;
Adaptive robust precision motion control of linear motors with negligible electrical dynamics: theory and experiments
Mechatronics, IEEE/ASME Transactions on , Volume: 6 Issue: 4 , Dec 2001
[7] Brogliato, B.; De Wit, C.C.; Villani, L.;
An exponentially stable adaptive control for force and position tracking of robot manipulators
Automatic Control, IEEE Transactions on , Volume: 44 Issue: 4 , Apr 1999

[8] Buchar, J.;Bartoň, S.; Křivánek I.:
Mechanika těles
Mendelova univerzita, Brno, 1996
[9] Corke, P.I.;
A robotics toolbox for MATLAB
IEEE Robotics & Automation Magazine , Volume: 3 Issue: 1 , Mar 1996
[10]Datta, A.; Ming-Tzu Ho;
A modified model reference adaptive control scheme for rigid robots
Robotics and Automation, IEEE Transactions on , Volume: 12 Issue: 3 , Jun 1996
[11] Dawson, D.M.; Dixon, W.E.; Zhang, F.; de Queiroz, M.S.;
Global adaptive output feedback tracking control of robot manipulators
Automatic Control, IEEE Transactions on , Volume: 45 Issue: 6 , Jun 2000
[12] Děmidovič, B.P.; Maron, J.A.:
Základy numerické matematiky
SNTL Praha, 1966
[13] Ferrarini, L.; Ferretti, G.; Maffezzoni, C.; Magnani, G.;
Hybrid modeling and simulation for the design of an advanced industrial robot controller
IEEE Robotics & Automation Magazine , Volume: 4 Issue: 2 , Jun 1997
[14] Fujimura, K.;
Path planning with multiple objectives
IEEE Robotics & Automation Magazine , Volume: 3 Issue: 1 , Mar 1996
[15] Gang Feng;
A new adaptive control algorithm for robot manipulators in task space
Robotics and Automation, IEEE Transactions on , Volume: 11 Issue: 3 , Jun 1995
[16] Gourdeau, R.;
Object-oriented programming for robotic manipulator simulation
IEEE Robotics & Automation Magazine , Volume: 4 Issue: 3 , Sep 1997
[17] Gu, E.Y.L.;
Configuration manifolds and their applications to robot dynamic modeling and control
Robotics and Automation, IEEE Transactions on , Volume: 16 Issue: 5 , Oct 2000
[18] Horský, J.;Novotný, J.; Štefaník, M.:
Mechanika ve fyzice
Academia, 2001
[19] Jing Yuan;
Adaptive control of robotic manipulators including motor dynamics
Robotics and Automation, IEEE Transactions on , Volume: 11 Issue: 4 , Aug 1995
[20] Kalaš, V.:
Mechatronické systémy s vysokou imunitou voči zmenám parametrov a zmenám záťaže
AT&P Journal, roč. VIII, plus 2, 2001
[21] Levine, W.:
The control Handbook
CRC Press, 1995
[22] Marhefka, D.W.; Orin, D.E.;
Animate: an educational tool for robot graphical simulation
IEEE Robotics & Automation Magazine , Volume: 3 Issue: 2 , Jun 1996
[23] Ming Liu;
Decentralized control of robot manipulators: nonlinear and adaptive approaches
Automatic Control, IEEE Transactions on , Volume: 44 Issue: 2 , Feb 1999
[24] Natale, C.; Koeppe, R.; Hirzinger, G
A systematic design procedure of force controllers for industrial robots

IEEE Trans. on Mechatronics, Vol.5.,Jun. 2000
[25] Nof, S.Y.:
Handbook of Industrial Robotics
J. Wiley,N.Y., 1985
[26] Pagilla, P.R.; Yu, B.; Pau, K.L.
Adaptive control of time-varying mechanical systems: analysis and experiments
IEEE Trans. on Mechatronics, Vol.5.,Dec. 2000
[27] Ranky, P.G.; Ho C.Y.:
Robot Modelling Control and Applications with Software
Springer Verlag, Berlin,1985.
[28] Rau, M.; Schroeder, D.:
Model Predictive Control with Nonlinear State Space Models
7th International Workshop on Advanced Motion Control, 3.-5.7. Univ. of Maribor, 2002
[29] Rojko, A.; Jezernik, K.:
Sliding Mode Motion Controller with Adaptive Fuzzy disturbance Estimation
7th International Workshop on Advanced Motion Control, 3.-5.7. Univ. of Maribor, 2002
[30] Sciavicco, L.; Siciliano, B.:
Modelling and Control of Robot Manipulators
Springer, 2002
[31] Snyder, W.E.:
Industrial Robots - Computer Interfacing and Control
Prentice Hall,N.J.,1985
[32] Spong, M.; Vidyasagar, M.:
Robot Dynamics and Control
J. Wiley,N.Y., 1989
[33] Stadler, W.:
Analytical Robotics and Mechatronics
McGraw-Hill, 1995
[34] Šolc, F.:
Řízení průmyslových robotů
ROBTEP'95, Prešov, 27.-29.9.1995
[35] Šolc, F.:
Matrix Model of Robot in Matlab – Simulink
Lecture Notes in Computer science
Springer Verlag, 2000
[36] Šolc F.:
Simulace pohybu průmyslového robota
XVI kolokvium, Vybrané problémy simulačních modelů 6.-8.9.1994Brno
MARQ Ostrava 1994
[37] Šolc, F.:
ASEA IRb6 Robot Calibration RAAD 2000
Int. Workshop, 1.-3.6.2000, Maribor
Univ.of Maribor,2000
[38] Šolc F.:
Dynamic Model of ASEA IRb6 Robot
XVII Workshop Advanced Simulation of Systems, 18.-20.4.1995
MARQ Ostrava 1995
[39] Šolc, F.:
CP Řízení a kalibrace robota ASEA IRb6
Nové trendy v strojárstve, 25.-26.9.1997, TU Prešov

[40] Tomei, P.;
Robust adaptive control of robots with arbitrary transient performance and disturbance attenuation
Automatic Control, IEEE Transactions on , Volume: 44 Issue: 3 , Mar 1999
[41] Vukobratovic, M.; Kircansky, M.:
Kinematics and Trajectory Synthesis of Manipulation Robots
Springer Verlag,1985
[42] Vukobratovic, M.:
Applied Dynamics of Manipulation Robots: Modelling, Analysis and Examples
Springer Verlag,1989

ABSTRACT
Habilitation thesis deals with modelling and control of stationary (industrial) robots. A matrix method of modelling of open kinematic chains together with electrical drives is developed and explained in the thesis together with several practical methods of control. All the described control schemes are tested by simulation with help of MATLAB-Simulink . Some of them were tested in practice.

ABSTRAKT
Habilitační práce se zabývá problematikou modelování a řízení stacionárních (průmyslových) robotů. V práci je uvedena maticová metoda modelování otevřených kinematických řetězců řízených elektrickými pohony. Současně jsou uvedeny a analyzovány některé praktické metody řízení takových systémů. Jednotlivé metody řízení jsou prověřeny simulací v prostředí MATLAB-Simulink. Některé metody byly ověřeny i v praxi.

ACKNOWLEDGEMENT
The author wants to thank to GACR which supported him by grants 101/93/2435 “Advanced schemes of robot control” and 102/02/0782 “Research in control of smart robotic actuators”