In this work a motion control scheme, which belongs to the class of the control schemes known as a sliding mode control with disturbance estimation is proposed. Adaptive fuzzy disturbance estimator works as an identifier of variable part of a system dynamics. Adaptation algorithm is derived by using the Lyapunov stability theory and provides global asymptotic stability of the state errors. For the implementation on the robot, physically based fuzzy logic subsystems are proposed. With this estimator’s transparency is enhanced and complexity reduced. If linguistic knowledge is available, subsystems enable its systematic inclusion. The control scheme was successfully tested on a laboratory direct drive (DD) robot.

1 Introduction

The purpose of this work was to develop a motion controller that can be used for DD robots. Their specific mechanical construction without gears results in complex and highly nonlinear dynamics. So for the control a model-free control is preferred. Sliding mode control, fuzzy control, neural networks and their combinations are between possible choices. Especially when combining sliding mode and fuzzy methods some very good motion controllers were developed (survey in [3]). For example, an adaptive fuzzy logic controller was used for approximation of the desired sliding mode control law with a boundary layer [1]. Controller based on fuzzy logic with an addition of discontinuous term of sliding mode control was described in [9]. Fuzzy logic was also used for tuning of the parameters of sliding mode controller [10]. However high complexity of most of those algorithms limits their applicability in practice.

In this paper proposed control method includes fuzzy logic for an estimation of almost whole system dynamics in the sliding mode control scheme. With use of the physically based subsystems it was possible to reduce rule base to only few rules, without influencing the estimation accuracy or robustness.

Paper is organized as follows. In the Section 2, a sliding mode control for DD robot is developed. An adaptive fuzzy logic system (FLS) for disturbance estimation is proposed in the section 3. In the Section 4, the validity of the proposed control scheme is verified by the experiments on the DD robot. The tracking accuracy for an average movement is shown. Summary of the work is given in the Section 5.
2 Sliding mode control design

Dynamics of \( k \)-th robot axis for a direct drive robot with \( m \)-degrees of freedom is:

\[
\tau_k = J_{ik}(q)\ddot{q}_k + \sum_{j=1}^{m} J_{kj}(q)\ddot{q}_j + \sum_{j=1}^{m} \sum_{l=1}^{m} C_{kl}(q)\dot{q}_l\dot{q}_j + G_i(q) + \tau_{s_k}(q,\dot{q}) + \tau_{d_k}
\]

(1)

where \( \tau_k \) is robot axis’s motor torque, \( J_i(q) \) is inertia, \( C_{kl}(q), G_i(q) \) are Coriolis and gravitation torques, \( \tau_{s_k}(q,\dot{q}) \) and \( \tau_{d_k} \) are friction and external disturbances torques. \( q_i, \dot{q}_i, \ddot{q}_i \) are axis position, velocity and acceleration. As disturbance let us define the whole dynamic effects (2), with exception of constant part of inertia.

\[
w_i = \Delta J_{ik}(q)\ddot{q}_i + \sum_{j=1}^{m} J_{kj}(q)\ddot{q}_j + \sum_{j=1}^{m} \sum_{l=1}^{m} C_{kl}(q)\dot{q}_l\dot{q}_j + G_i(q) + \tau_{s_k}(q,\dot{q}) + \tau_{d_k} + \tau_{d_k}
\]

(2)

Now the dynamics (1) can be rewritten as:

\[
\tau_k = J_{ik}q_{ik} + w_i(q,\dot{q},\ddot{q}),
\]

(3)

where \( J_{ik} \) denotes the constant part of inertia.

For the control design we employ a decentralized switching scheme. To include tracking requirements, all \( m \) sliding surfaces have to be chosen as a function of acceleration, velocity and position errors:

\[
\sigma_k = (\ddot{q}_i^d - \ddot{q}_i) + g_{1,k}(\dot{q}_i^d - \dot{q}_i) + g_{2,k}(q_i^d - q_i),
\]

(4)

where \( g_{1,k} \) and \( g_{2,k} \) are positive constants and superscript \( d \) denotes reference values. Because this switching function is a function of unknown actual acceleration, it cannot be calculated. To solve this problem the method from [2] can be used. This method proposes a transformation of the switching functions, so that no actual acceleration is needed. Transformed switching functions still define the same sliding manifolds as original switching functions. Let the transformed switching functions be:

\[
\sigma_{transf,k} = i_{i,k}^\prime - i_{i,k},
\]

(5)

with reference current \( i_{i,k} \) and actual current are defined as \( i_{i,k} \):

\[
i_{i,k} = \frac{J_{ik}}{K_{M,k}} \dot{q}_i + \frac{1}{K_{M,k}} w_i, \quad i_{i,k}^\prime = \frac{J_{ik}}{K_{M,k}} \dot{q}_i^\prime + \frac{1}{K_{M,k}} \hat{w}_i
\]

(6)

In actual current is actual acceleration replaced by calculated one \( \ddot{q}_i^\prime \) (7), derived from the condition of sliding regime \( \sigma_k = 0 \). Also the disturbance is in actual current replaced by estimated one \( \hat{w} \).
\[
\sigma_i = 0 \Rightarrow \ddot{q}_i^n = \ddot{q}_i^k + g_{2,i}(\dddot{q}_i^k - \dot{q}_i^k) + g_{1,i}(\dot{q}_i^k - q_i^k)
\]

If we now rewrite (5) with using (6) we get:

\[
\sigma_{\text{transf}} = \sigma_i^k + \frac{1}{K_{M,k}}(\ddot{w}_k - w_k).
\]

From (8) can be seen that when \(\lim_{w_k \to 0} (\ddot{w}_k - w_k) = 0\), \(k=1..m\), the transformed switching functions define the same sliding manifolds as original switching functions. The importance of good disturbance estimation can be also shown by error equation on \(\sigma_{\text{transf}} = 0\).

\[
[\ddot{q}_i^k - \dot{q}_i^k] + g_{2,i}(\dddot{q}_i^k - \dot{q}_i^k) + g_{1,i}(\dot{q}_i^k - q_i^k) = J_{ii}^k[\dot{v}_i - w_i]
\]

For disturbance estimation a lot of different schemes can be used. Some of them were tested on our three degree of freedom direct drive robot, but failed to provide a good tracking accuracy specially for high speed movements. So in the next section of the paper an emphasis is put on a design of a good disturbance estimator.

### 3 FLS for disturbance estimation

One multiple input - single output FLS for a perturbation estimation (under perturbation we consider all dynamic effects with exception of the constant part of inertia) was designed for each robot axis. A vector of FLS input variables is defined as \(x_i = [\ddot{q}_i^k, \dot{q}_i^k, \dot{q}_i^k]^T\), where \(\dddot{q}_i^k\) is desired acceleration used instead of an unknown actual acceleration. Fuzzy rule base of FLS on \(k\)-th robot axis consist of IF-THEN rules \(R_i^l\) in the following form:

\[R_i^l : \text{if } \dddot{q}_i^k = X_i^{k,l} \text{ and } \dot{q}_i^k = X_i^{k,l} \text{ and } \ddot{q}_i^k = X_i^{k,l} \text{ then } \dddot{w}_i = \dddot{r}_i
\]

Superscript \(l\) refers to the \(l\)-th rule \(l=1..M\). \(X_i^{k,l}, X_i^{k,l}, X_i^{k,l}\) are fuzzy sets in an input universe of discourse, \(\dddot{w}_i\) are output linguistic variables and \(\dddot{r}_i\) are singleton fuzzy sets in the output universe of discourse.

The structure of FLS was chosen as: singleton output membership functions, singleton fuzzifier, product-operation rule of fuzzy implication, sum-operation for fuzzy aggregation and center of average deffuzifier. Bell shaped function form was chosen for input membership functions. The output of the resulting FLS can be calculated as:
where $\tau_i$ are the centers of the input membership functions, $\pi_i$ are positions of the output membership functions, $\sigma_i$ determine the width of the bell function and $\beta_i$ its slope. $x_i$ refers to the i-th element of the vector of input variables. FLS in the form of (11) are universal approximators, capable to approximate any nonlinear continuous function on a compact set to arbitrary accuracy, [8].

After choosing the structure of FLS its parameters have to be determined. To find suitable parameters of input membership functions is an easy task. First we set their number and then determine their positions, widths and slopes, so that they cover all possible values of the input variables. These values are physically limited by the positions of the limit switches and by the maximal motor torques. More pretentious is to set the positions of the output membership functions. First, there is no sufficient linguistic knowledge for this. Second, by fixing them, the tracking accuracy would decay rapidly in the presence of unpredictable dynamic changes such as varying payload. So the positions of the singleton output membership functions $l_k$ have to be adaptive.

Let us put all adjustable parameters into parameter vector $\theta = [\theta_1, \ldots, \theta_n]$ and the rest of the expression (11) into a vector of known nonlinear functions $\xi_i(x_k)=[\xi_i^1(x_k), \ldots, \xi_i^m(x_k)]$, where $\xi_i^j(x_k)$ are defined with (12).

$$\xi_i = \left(\prod_{i=1}^{n} \mu_{\pi_i}(x_i)\right) \div \left(\sum_{i=1}^{n} \prod_{i=1}^{n} \mu_{\pi_i}(x_i)\right).$$

(12)

With using these two vectors, the FLS (11) can be written in the parameter vector – regressor form (13), known from the classical theory of system identification [4].

$$\dot{\theta}_i = \dot{\theta}_I \cdot \xi_i(x_k)$$

(13)

Next the adaptation law has to be determined, so that the difference between the optimal parameter vector $\theta_i^*$ (which guarantee perfect perturbance estimation) and used estimated parameter vector $\hat{\theta}_i$, which is denoted by $\tilde{\theta}_i$, is minimized:

$$\Delta \theta_i = \dot{\theta}_i - \hat{\theta}_i = \left(\hat{\theta}_I - \theta_I^*\right) \cdot \xi_i(x_k) = \tilde{\theta}_I \cdot \xi_i(x_k).$$

(14)

The adaptation law must assure a global asymptotic stability of equilibrium point $e_i=[e_i, e_i] = 0$ and can be derived by using Lyapunov stability theory. Let us chose following Lyapunov function:
where $A_k$ is a symmetric, positive definite. It can be proven, that the derivative of this Lyapunov function is negative semi-definite and therefore equilibrium point stable, if we use adaptation law (16). Further it can be shown by using the Barbalat theorem, that the equilibrium point is asymptotically stable (details in [5]).

In the derivation process it was assumed that $\dot{\theta} = \dot{\hat{\theta}}$. This is a usual in the design of adaptive systems and it means that the optimal parameter vector is constant. However the derived adaptation law can be also used on the systems with time varying parameters, if the adaptation is much faster than parameter changes.

Next we try to reduce the complexity of FLS. FLS on each robot axis was divided to three FLSB, each for estimation of one part of dynamic influencing this robot axis. First FLSB is designed to estimate the difference between the average inertia, which is already used in the control scheme, and the actual inertia. Its inputs are position and desired acceleration, $x_k = [\theta_k, d\theta_k]$.  

$$\hat{\omega}_{1,FLS} = (J_{\hat{\theta}}(\theta_k) - J_{\theta_k}) \cdot \ddot{\theta}_k. \quad (17)$$

Second FLSB inputs are position and velocity, $x_k = [\theta_k, \dot{\theta}_k]$, and it is designed for an estimation of the gravitation, Coriollis, centrifugal forces and velocity and position dependent friction effects:

$$\hat{\omega}_{2,FLS} = C(q_k, \dot{q}_k) + G(q_k) + \tau_{e_k} \cdot (\dot{q}_k, \ddot{q}_k). \quad (18)$$

Third FLSB is designed for the estimation of other effect and eventual residue of dynamics that should be estimated by the first two FLSB. Its inputs are actual position, velocity and desired acceleration, $x_k = [\theta_k, \dot{\theta}_k, d\theta_k]$.  

Estimator’s transparency is by using FLSB preserved, regardless of the complexity of its task (estimating the dynamic of a direct drive robot). Also the number of inputs into single rule is reduced. Additionally the decision which rule of all possible rules should be included in the rule base is much easier. This all together reduces the complexity of FLS without negatively influencing its performance and enables a real-time implementation of the controller.

### 4 Application to motion control of direct drive robot

Our control plant is a three-degree of freedom Puma-like configuration direct drive robot, Figure 1.
Three membership functions have been used for each of the inputs of FLS, same for all three axes, Figure 2. All used rules are shown in Table 1. Sliding manifold parameters were chosen as $g_{1,x}=1000$, $g_{1,y}=2400$, $g_{1,z}=1200$, $g_{2,x}=63.25$, $g_{2,y}=97.97$, $g_{2,z}=69.28$, parameters of diagonal inertia matrix as $J_{1}=3.5$, $J_{2}=2.5$, $J_{3}=0.13$ kgm². Learning parameters were $a_{1,k}=220$, $a_{2,k}=1.5$, $a_{3,k}=250$, $a_{4,k}=5$, $a_{5,k}=35$, $a_{6,k}=1$ and the learning rates as $\alpha_{a_{i,k}}=0.2$.

Experimental results are shown for an average movement. A reference trajectory was a point-to-point movement; same for all three axes. Reference position and velocity of robot tip is shown in Figure 3. The resulting robot tip’s position error with a peak error of 2.6 mm and zero positioning error is shown in Figure 4. Figure 5. shows nominal torques and torques form FLS for all robots’ axes for this movement. On second and third axis most of the torque origins from FLS, which confirms the quality of disturbance estimation.

5 Conclusion

In this paper the development and implementation of a motion controller for a direct drive robot has been presented. Sliding mode control with fuzzy disturbance estimation has been used. To cope with highly non-linear dynamics of robot manipulator and incomplete linguistic knowledge, an on-line adjustment of some FLS’s parameters has been used. The control scheme is completely decentralized and besides some basic linguistic knowledge requires only known average inertia matrix. Its simple structure and small number of linguistic rules enable easy implementation of proposed control on almost every controller hardware.

Figure 1. Control object, direct drive robot
Figure 2. Input membership functions

<table>
<thead>
<tr>
<th>RULE</th>
<th>POSITION</th>
<th>VELOCITY</th>
<th>ACCELERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. subsystem</td>
<td>R¹ negative</td>
<td>-</td>
<td>negative</td>
</tr>
<tr>
<td></td>
<td>R² zero</td>
<td>-</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>R³ positive</td>
<td>-</td>
<td>positive</td>
</tr>
<tr>
<td>2. subsystem</td>
<td>R⁴ negative</td>
<td>negative</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>R⁵ zero</td>
<td>zero</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>R⁶ positive</td>
<td>positive</td>
<td>-</td>
</tr>
<tr>
<td>3. subsystem</td>
<td>R⁷ positive</td>
<td>zero</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>R⁸ negative</td>
<td>zero</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>R⁹ zero</td>
<td>negative</td>
<td>positive</td>
</tr>
<tr>
<td></td>
<td>R¹⁰ zero</td>
<td>zero</td>
<td>zero</td>
</tr>
<tr>
<td></td>
<td>R¹¹ zero</td>
<td>zero</td>
<td>positive</td>
</tr>
<tr>
<td></td>
<td>R¹² zero</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td>Not implemented on 1. axis</td>
<td>R¹³ positive</td>
<td>zero</td>
<td>positive</td>
</tr>
<tr>
<td></td>
<td>R¹⁴ negative</td>
<td>negative</td>
<td>negative</td>
</tr>
<tr>
<td></td>
<td>R¹⁵ positive</td>
<td>positive</td>
<td>positive</td>
</tr>
</tbody>
</table>

Table 1: Rules in FLS

Figure 3. Reference trajectory
Figure 4. Robot tip’s position error and applied torques

References