Coordinated Control of Hydraulic Mobile Manipulators

Reza Ghabcheloo, Kalevi Huhtala

* Department of Intelligent Hydraulics and Automation, Tampere University of Technology, Tampere, Finland (e-mail: reza.ghabcheloo@tut.fi, Kalevi.huhtala@tut.fi).

Abstract: In this paper, we propose a solution to the problem of path following for a hydraulic mobile manipulator. First, we provide a solution to the trajectory generation for each joint given desired path of the work tool. The solution is based on forward kinematics and stability theory of dynamics systems. Avoiding inverse kinematics reduces computational effort and complexity. We then introduce a feedback control strategy that makes the work tool follow the desired trajectory. The solution is so called time-coordinated path following (TC-PF). Firstly, the solution relies on path-following instead of trajectory tracking, thus avoiding performance limitations inherited by trajectory tracking. Secondly, a coordination algorithm is added to the feedback system that synchronizes all joints guaranteeing that the work tool stay in the neighborhood of the desired path even if a joint fails to follow its trajectory. In other words, if one of joints exhibits slow dynamics, all other joins will slow down in such a way to keep the work tool on the path. Similar idea can be extended to operation of multiple mobile manipulators in close vicinity.

Keywords: Path following, Hydrostatic manipulators, Forward kinematics path generation.

1. INTRODUCTION

A robotic system in general is composed of three main subsystems: sensing, planning, and control. The most complex control architecture may include nine levels in hierarchical configuration (Siciliano et al. (2008)); however they all include the three parts mentioned above. The emphasis of this paper is on control for hydraulic manipulators. Many tasks that robotic manipulators are required to perform include transition of the tool / hand / end-effector from an initial configuration to a goal configuration, while avoiding possible obstacles. Tasks are naturally given in work space or operational space usually a Cartesian space, and planner’s job is to generate sequence of way points that lead the end-effector to the goal. Since the control actions are done in joint space, there is always a need for conversion of plans from work space to joint space or vice versa. Traditional control architectures (Klafter et al.(1989)) are

- JPS: Joint position control
- RMPC: Resolved motion position control
- RMRC: Resolved motion rate control

First method is only suitable for control of a single joint at a time. RMPC is essentially the same as JPS. The main difference is that RMPC employs inverse kinematics; therefore plans can be generated in work space. Both RMPC and JPS are essentially open-loop controls. In contrast, RMRC has a closed-loop structure, and can be used in more complex and uncertain environment. In this paper, we propose a new implementation methodology, so called Time-Coordinated Path Following, TC-PF in short. This method was first introduced in Ghabcheloo et al. (2009 c). TC-PF for mobile manipulators is a variation of JPS in which trajectory tracking has been replaced by path following, and evolution of the target point on the path is manipulated in such a way to reduce end-effector’s tracking error. Proposed TC-PF solves the problems that are inherited in JPS. That is, a feedback structure will reduce tracking error and keep the joints synchronized.

Based on the results in Galicki (2004), we will also propose a joint trajectory generation method that does not require inverse kinematics, and accommodates actuator limits (in joint space) and other obstacles (in work space).

Fig. 1 shows how time-coordinated path following is implemented. TC-PF is composed of three main subsystems: I) trajectory generator which is described in Section 2; II) a speed servo for each joint, see Jelali et al (2003), and Ghabcheloo et al. (2009 a); and III) path following and coordination control that synchronize the joints; this is elaborated in Section 3. For simplicity of the presentation, we will derive the equations for an exemplar hydraulic mobile manipulator, namely an articulated-frame steered fork-lift as shown in Fig. 2. The machine is hereafter called GIM mobile machine. Section 4 provides an example where we show how TC-PF is implemented on GIM mobile machine, driving it from an initial configuration to the pallet. Section 5 touches the extension of the proposed method to multiple machines.

Given initial and final goal configuration of the tool, trajectory planner generates a time profile for each joint $q_d(t)$, where $d$ stands for desired values, subscript $i \in \{1, \ldots, n\}$ denotes the joint index with $n$ the number of the joints. To distinguish real time that unfolds during execution of the plans, we replace variable $t$ in the argument of $q_d(t)$ by a new variable $\gamma$, so called virtual time, evolution of which is manipulated to guarantee small tracking error. If everything goes as planned, we will recover $\dot{\gamma} = 1$. Benefit of this method is more pronounced when the system is composed of actuators with different dynamics. Tracking bandwidth is limited by the slowest joint speed servo. TC-PC reduces the speed of the evolution of the trajectory to accommodate for the slowest joint, so that the end-effector remains in the vicinity of the desired path in the work space.
2. JOINT SPACE TRAJECTORY GENERATION

The aim of this paper is to derive control algorithms that enable GIM mobile machine move towards a pallet and enter its fork tool into the pallet without collision, as shown in Fig 2. We will assume that an independent steering feedback controller keeps the machine in front of the pallet, this is addressed in Ghabcheloo et al. (2009 a, b). Therefore, the problem will be addressed only in two dimensions, that is, forward and downward, \( x - z \).

Let column vector \( \mathbf{q} = (x \ l \ \theta_1 \ \theta_2)^T \) denote the generalized coordinates of the mobile manipulator, or the joint variables, and \( P(\mathbf{q}) \) and \( \Phi(\mathbf{q}) \) denote the position and orientation of the end effector, respectively, in the work space (pallet frame in this example). Variable \( x \) is the distance of the machine to the pallet, \( l \) and \( \theta_1 \) are length and angle of the boom, and \( \theta_2 \) is the angle of the fork. Exact definitions of these variables will be given in Section 4. We assume all variables are independently controllable and there exist speed servos for each control variable, that is, four single-input single output speed tracking servo control for \( x \ l \ \theta_1 \) and \( \theta_2 \). There are two other important issues worth mentioning at this point: a) in the case of GIM mobile machine, servo actuators exhibit significantly different dynamics, since the masses are different and \( x \) is controlled by a pump, and the rest are control by proportional valves; b) GIM mobile machine is a redundant manipulator (moving the fork frame forward can be made either by controlling the body or the work hydraulics).

We further assume that there is a planner that generates a path for the fork in work space from the initial configuration to the goal configuration: a desired path \( \mathbf{p}(s) \) and orientation \( \Phi(s) \) is then given in the work space parameterized by \( s \in [0, s_f] \), an arbitrary variable. For example, \( s \) can be time, or length of the path measured from an initial point. The objective is to find joint space variables that correspond to the desired work space variables. In other words, we would like to generate time trajectories for \( x, l, \theta_1, \) and \( \theta_2 \) in such a way to keep \( P(\mathbf{q}) \) close to \( \rho(s) \), \( \Phi(\mathbf{q}) \) close to \( \Phi(s) \). This is, in a sense, an inverse kinematic problem (Scicolano et al. (2010)). Next we present a solution to this problem without a need for inverse kinematic. The solution employs theorems from stability and control theory.

Path following error at time \( t \) is defined as

\[
\begin{align*}
e_p(t) &= P(q(t)) - \rho(s(t)), & \text{position error} \\
e_o(t) &= \Phi(q(t)) - \Phi(s(t)), & \text{orientation error}
\end{align*}
\]

Following proposition provides tools for us to derive dynamic equations that generate trajectories for coordinates \( \mathbf{q} \) such that the path following errors for the end effector remain small, while \( \dot{q} \) is kept bounded at will. We let \( U(q, s) \) denote an obstacle avoidance function (a non-negative smooth function), that is, \( U(q, s) = 0 \), when the manipulator is in safe distance from the obstacles, and \( U(q, s) \) takes large values, when the manipulator gets too close to the obstacles.

**Proposition 1.** Let \( q_d(t), s_d(t) \) be the solution of

\[
\begin{align*}
\dot{q} &= K_q^{-1} \left(-k_1 \dot{q} - \beta_1 \frac{\partial \Phi}{\partial q} \dot{q} - \beta_2 \frac{\partial \Phi}{\partial q} \dot{q} - \beta_3 \frac{\partial \Phi}{\partial q} \dot{q} \right) \\
\dot{s} &= -k_2 \dot{s} + \beta_3 \frac{\partial \Phi}{\partial s} \dot{s} + \beta_2 \frac{\partial \Phi}{\partial s} \dot{s} + \beta_1 \frac{\partial \Phi}{\partial s} \dot{s} + \gamma_s(s_o - s) \\
\end{align*}
\]

with initial condition \( s_d(0) = 0, \dot{s}_d(0) = 0, \dot{q}_d(0) = 0 \), and a \( q_d(0) \) that satisfies \( P(q_d(0)) = \rho(0) \), and \( \Phi(q_d(0)) = \Phi(0) \), where \( k_1, k_2, \beta_1, \beta_2, \gamma_s \) and \( K_q \) are positive variables. Then for all time,

\[
\begin{align*}
|K_q q_d(t)| &\leq \gamma_s, \ |e_p(t)| \leq \frac{\gamma_0}{\sqrt{\beta_1}}, \ |e_o(t)| \leq \frac{\gamma_0}{\sqrt{\beta_2}} \\
\end{align*}
\]

Moreover, in the absence of obstacles, that is, when \( U(q, s) = 0 \) variables \( \dot{q}, \dot{s} \) asymptotically converge to zero, that is, for any desired tolerance value \( \epsilon \), there is final time \( t_f > 0 \), such that all the errors become smaller than \( \epsilon \) after time \( t_f \), in particular

\[
\begin{align*}
|q_d(t_f)| &< \epsilon, \\
|s_d(t_f) - s_f| &< \epsilon
\end{align*}
\]

Proof follows similar steps as in Galicki (2004).

In words, Proposition 1 states that given desired path and orientation for the end effector, solution of (1) provides desired trajectories for each joint without resorting to inverse kinematic. Notice that the solution is rather general, and all we need is to calculate forward kinematics and the Jacobians for the given mobile manipulator. Error tolerances can be reduced by increasing \( \beta_1, \beta_2 \), see (2). Moreover, from (2), it is easy to see that speed of the joints can be limited using \( \gamma_o \), while \( K_q \) can be used to normalize or equally scale different joints speed. Equation (3) and initial condition show that the mobile manipulator will start from a stand still state and will come to almost stop defined by \( \epsilon \) at the end of the trajectory.
For each joint, solution of (1) will generate generalized desired trajectory $Q_d: (q_d(y), q_d'(y))$, where prime sign ' stands for derivative with respect to $y$. We used the term generalized trajectory because $Q_d$ includes both the trajectories and the derivatives. Solution is calculated till bounds in (3) are satisfied. We then have $y \in [0, t_f]$. In the next section, we will elaborate time-coordinated path following strategy, a feedback control that make each joint follow assigned trajectories given by $Q_d$ in harmony with the other joints in such a way to keep the end effector close to the planned trajectory.

3. TIME-COORDINATED PATH FOLLOWING FOR A MOBILE MANIPULATOR

Consider a hydraulic mobile manipulator in which speed of each joint is controlled independently, that is, joint control systems include speed servos. This is usually the case, since most of the hydraulic actuators are controlled by valves or pumps where flow is controlled and effect of load pressure is compensated. A good low frequency approximate model of pumps where flow is controlled and effect of load pressure is negligible. Consequently, path following errors $e_p$ and $e_q$ will remain small. Before further developments, we need to introduce matrix $L$, a positive semi-definite symmetric matrix. Its rank equals $n - 1$ and $L[1] = [0]_n$, where column vectors $[1]_n$ and $[0]_n$ are of dimension $n$ with all elements equal 1 and 0, respectively. Furthermore, we stack all the virtual time variables into $y = [y_i]_{i=1..n}$. Properties of $L$ guarantee that $y^T Ly$ can be written as sum of terms like $(y_i - y_j)^2$. See Ghabcheloo et.al (2009 c) for details.

Proposition 2. Consider the system of equations (4) together with $n$ variables $y_i$. Let the evolution of the virtual time variables be governed by

$$\dot{y}_i = v_i - k_p \max(0, y_i - q_d(y_i))$$

where $v_i$ is given by (7), and $k_p, \alpha_i$ and $\beta_i$ are positive constants, and $L_i$ is the $i$'th row of $L$. Moreover, let the control signals be given by

$$u_i = q_d'(y_i)v_i - k_p \max(0, y_i - q_d(y_i))$$

Then, tracking errors $y_i - q_d(y_i)$ and coordination errors $y_i - y_j$ will remain negligible if the errors are initially zero. Furthermore, the control system is robust in the sense that the errors remain small in either of the following non-ideal cases:

- In the presence of bounded disturbances that can be modelled as $\dot{q}_i = u_i + d_i$, with $|d_i| < \delta$. The error bounds can be made small with larger $k_p$,.

- If any of the actuators fails, say $\dot{q}_j = 0$ for some $j$.

Proof. Define the following Lyapunov function that includes the path following errors and coordination errors,

$$V = \frac{1}{2} \sum_i \beta_i (y_i - q_d(y_i))^2 + \frac{\alpha}{2} y^T Ly.$$ 

Notice that $V = 0$ if and only if $q_i = q_d(y_i)$ and $y_i = y_j$. The derivative of $V$ yields

$$\dot{V} = \sum_i \beta_i (q_i - q_d(y_i))(u_i - q_d'(y_i)) + \alpha y^T Ly$$

$$= \sum_i \beta_i (q_i - q_d(y_i))u_i + (\alpha L y - \beta_i (q_i - q_d(y_i)) q_d'(y_i)) y_i$$
Substituting \( u_i \) from (9) and \( \dot{y}_i \) from (8) results in
\[
\dot{y} = - \sum_k \beta_k p(q_i - q_{di}(\gamma_i))^2 + k_g \left( a_L \gamma_i - \beta_i (q_i - q_{di}(\gamma_i)) q_{di}'(\gamma_i) \right)^2
\]
which is negative definite, thus \( V \) and consequently the path following errors will remain negligible if \( V \) is initially zero.

If one of the joints stops, say \( \dot{y}_j = 0 \), then \( \dot{V} \) renders
\[
\dot{V} = -v_d \beta_p q_{di}'(q_j - q_{di}(\gamma_j)) - \sum_k \beta_k p(q_i - q_{di}(\gamma_i))^2
\]
\[
- \sum_k k_g \left( a_L \gamma_i - \beta_i (q_i - q_{di}(\gamma_i)) q_{di}'(\gamma_i) \right)^2
\]
As it is seen, \( \dot{V} \) is not necessarily negative. However, if the errors grow above the limit values, \( v_d \) thus the indefinite term vanishes according to (7), and \( \dot{V} \) remains non-positive, thus errors remain bounded.

Now let analyse the effect of bounded disturbances. Let \( \dot{q}_i = u_i + d_i \), with \( |d_i| < a \). Then
\[
\dot{V} = \sum_k \beta_k d_i(q_i - q_{di}(\gamma_i))
\]
\[
- \sum_k \beta_i (k_p - \alpha_i) (q_i - q_{di}(\gamma_i))^2
\]
\[
+ k_g \left( a_L \gamma_i - \beta_i (q_i - q_{di}(\gamma_i)) q_{di}'(\gamma_i) \right)^2
\]
After some simplification, we can write
\[
\dot{V} \leq \sum_k \frac{1}{k_d_i} \beta_k d_i^2
\]
\[
- \sum_k \beta_i (k_p - \alpha_i) (q_i - q_{di}(\gamma_i))^2
\]
\[
+ k_g \left( a_L \gamma_i - \beta_i (q_i - q_{di}(\gamma_i)) q_{di}'(\gamma_i) \right)^2
\]
Therefore, for \( \alpha_i < k_p \), the system is ultimately bounded. Clearly the bound is smaller, for larger \( k_p \).

Notice that the control law given by (9) is similar to (6), except that in (9) desired trajectory for each joint \( q_{di}(\gamma_i) \) is generated in its own pace, that is, by its own virtual time \( \gamma_i \). Furthermore, (8) guarantee that the difference between virtual time variables remain small so that the end effector remains close to the desired path. Intuitively, when a joint fails in tracking its assign trajectory, term \( (q_i - q_{di}(\gamma_i)) q_{di}'(\gamma_i) \) in (8) will keep \( \gamma_i \) of growing further. And term \( L_i \gamma_i \) in (8) guarantees that errors \( \gamma_i - \gamma_j \) do not grow, thus all the other joints also slow down. Furthermore, in ideal case when all joints follow closely their corresponding trajectory, \( \gamma_i = 1; \forall i \), and trajectories evolve as fast as real time.

According to Proposition 2, in the presence of disturbances, tracking errors can be made smaller by increasing \( k_p \). Unfortunately, \( k_p \) cannot be made too large, and it is limited by the bandwidth of the speed servo (the inner loop), in other words, limited by stability gain margin of the control loop. Thus, better disturbance rejection properties are achieved by higher bandwidth of joint speed servos.

It is shown in the proof that when one of the joints stops, path following errors may grow to a limit defined by \( \delta \), then \( v_d \) becomes zero, which in turn causes the virtual time variables to stop. That is, all the joints will stop after a transient, thus the end effector will remain on the vicinity of the assigned path, but will not move farther.

4. ILLUSTRATIVE EXAMPLE

Let \( \{P\}, \{B\}, \) and \( \{F\} \) be coordinate frames attached to the pallet, the body, and the fork, respectively. See Fig 2. We let the work space coordinate system be attached to \( \{P\} \) which is an earth fixed frame. Let \( P_A \) denote the position of point \( A \) in frame \( \{P\} \), \( O_B \) denote the origin of frame \( \{B\} \), \( p_R \) rotation matrix from \( \{B\} \) to \( \{P\} \). Other positions and rotations are defined in similar manner. Position of the end effector in the pallet frame is given by
\[
^P P_{OF} = ^P P_{OB} + ^P R_A A^R d_P
\]
where \( ^P A = (d \ h)^T \), \( ^P P_{OB} = (x \ z)^T \). Here, we assume that the machine moves on a horizontal plane with respect to \( \{P\} \) frame, thus \( d, h, z \) are constant variables, and \( p_R \) and \( A^R \) are identity. Then \( ^P P_{OF} \) simplifies to
\[
^P P_{OF} = P(q) = \left( \begin{array}{c} d + x \cos \theta_1 \\ h + z \sin \theta_1 \end{array} \right).
\]

Orientation of the end effector in \( \{P\} \) is given by \( \Phi(q) = \theta_2 \).

4.1 Joint variables trajectory generation

Suppose we would like to drive the end effector from the following initial configuration
\[
(x(0), l(0), \theta_1(0), \theta_2(0)) = (-3.1m, +1.8m, -0.35rad, +0.35rad)
\]
to the pallet (the origin). Let the desired path \( \rho(s) \) and \( \phi(s) \) be given by
\[
\rho(s) = \left( 1 - \frac{s}{s_{mid}} \right) \left( \begin{array}{c} d + x(0) \cos \theta_1(0) \\ h + z - l(0) \sin \theta_1(0) \end{array} \right) + \frac{s}{s_{mid}} \left( \begin{array}{c} d \\ h \end{array} \right)
\]
\[
\phi(s) = \left( 1 - \frac{s}{s_{mid}} \right) \theta_2(0)
\]
for \( 0 \leq s < s_{mid} = \frac{s_f}{4} \) and
\[
\rho(s) = \frac{s_f - s}{s_f - s_{mid}} \left( \begin{array}{c} d \\ h \end{array} \right)
\]
\[
\phi(s) = \frac{s_f - s}{s_f - s_{mid}} \theta_2(0)
\]
for \( s_{\text{mid}} \leq s < s_f = 1 \). Notice that the path starts from initial condition and ends at the pallet, and consists of two pieces of straight lines. This path example was only presented for the sake of completeness. Usually elaborated planning algorithms are needed, see Siciliano et al (2010).

Function \( U(q, \eta) \) can be used to prevent generating trajectories that collide with the pallet, or hit the fork to the ground, or exceed the limits of the telescopic boom. Suppose, we would like to limit \( l \in (l_{\text{min}}, l_{\text{max}}) \) with a safe distance \( l_{\text{safe}} \). Galicki (2004) suggest functions of the following form

\[
U = \begin{cases} 
  w e^4 & e \leq 0; e := l - (l_{\text{min}} + l_{\text{safe}}) \\
  w e^4 & e \geq 0; e := l - (l_{\text{max}} - l_{\text{safe}}) \\
  0 & \text{otherwise}
\end{cases}
\]

(13)

where \( e \) is a distance function, and \( w \) a positive number. However, functions such as in (13) result in fast oscillatory behaviour when the actuators reach to end. To address this issue, we suggest an obstacle function that includes a damping factor: \( \frac{d}{dt} e = w^2 e + 2w \ddot{q}_i \), with \( w = \frac{1}{l_{\text{safe}}} \) and \( e \) is defined similar as in (13). Notice than in (1), we only need derivative of \( U \), and not \( U \) itself.

Due to redundancy, the solutions may be undesirable. To generate more predictable outcomes, we would like to keep the fork close to the machine when the machine is far from the pallet. We then let

\[
U = w (d + l \cos \theta_1 - 0.5)^2 \quad \text{with} \quad w = 10
\]

This will penalize horizontal distance of \( \text{O}_F \) to \( O_H \), calculated by \( d + l \cos \theta_1 \), and keep it close to 0.5m.

Since the solution of (1) is only asymptotically convergent, speeds are high in departure and become smaller as time increases. Thus convergence becomes very slow when \( s \) approaches \( s_f \). Therefore solution can be unnecessarily long and slow when arriving close to the pallet. To speed up convergences, we can set the target \( s_f \) farther than terminal \( s_f \). The following table shows terminal time \( t_f \) versus target terminal \( s_f \), where simulation is terminated at \( s = 0.99 s_f \).

<table>
<thead>
<tr>
<th>Final time[s]</th>
<th>42.6</th>
<th>13.8</th>
<th>13.7</th>
<th>15.2</th>
<th>18.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_f )</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

Generated trajectories are shown in Fig. 3 for three values of \( s_f \). Red dashed lines indicate limit of the actuators. From the most left graph in the figure, we can see that machine will move forward till about 0.8m to the pallet, then the body stops and telescopic boom extends to reach and end the pallet. Boom and fork angles should correspondingly change to keep the fork on the desired path. Fig. 4 shows resultant motion of the machine in the work space coordinates. Control parameters were set as follows

\[
\gamma_0 = 2, \beta_1 = 100, \beta_2 = 100, \beta_3 = 10, k_2 = 2.7, \\
k_1 = \frac{10}{k_2} \text{diag}(0.5, 2, 2, 2), K_q = \text{diag}(0.5, 0.4, 0.5, 1)
\]

Moreover the dimension parameters of the GIM machine are

\[
d = -0.59[\text{m}], h = -0.94[\text{m}], z = -0.17[\text{m}], \quad \text{and} \quad f = 0.85[\text{m}].
\]

4.2 Time coordinated path following

In this section, we evaluate the concept of time-coordinated path following introduced in the paper using a simulator. In the simulator bandwidth of the speed servos for each channel are roughly as follows:

Drive line \( BD_1 = 0.4\text{Hz} \),
boom telescope \( BD_1 = 0.8\text{Hz} \),
boom angle \( BD_\theta_1 = 0.8\text{Hz} \),
fork angle \( BD_\theta_2 = 1.5\text{Hz} \). We choose the planned trajectories shown in Fig. 3 for \( s_f = 2 \).

Final time of the generated planned is 13.7s. Because the simulated machine differs from the ideal low frequency approximation (4), end effector will deviate from the planned path.

To compare different controllers, we define the following scalar value as path following error

\[
e = e^T p e + e^T \dot{q}_i
\]

(14)

which is sum of square of all path following errors. Fig. 6 shows time trajectory of error \( e \) for four experiments

a) Left-top: Pure trajectory tracking: control law (5)
Notice that pure trajectory tracking, case a, has the poorest transient response. Adaptation of $v_d$, case b, considerably reduces the error in the cost of increased execution time. In the latter case, because the joints cannot follow closely enough the desired trajectory, $v_d$ is reduced according to (7) and the virtual time grows slower than the real time. The best result is achieved in experiment c, where path following control with both adaptive $v_d$ and coordinated virtual time variables is applied. In this case, the error level is the smallest and yet final time is not as large as in experiment b. Experiment c with $v_d = 1$ was as poor as experiment a, concluding that only TC-PF with no adaptation of $v_d$ is not enough to improve the results. Fig. 7 shows how virtual time variables grow in experiment c. Small deviations from the plan can be seen by comparing Fig. 4 and Fig. 5. Notice that the snapshots are taken every 2 seconds.

5. MULTI-MACHINE CASE

TC-PF strategy can be extended to be used in planning and control of multiple mobile manipulators that work in the vicinity of each other, where synchronization and fluent work is essential. In this case, vector $q$ will consist of the joint states of the entire fleet, and matrix $L$ will be a function of Laplacian matrix of the communication network. Effects of network connectivity, delays, and communication losses will then need to be addressed.

6. CONCLUSIONS

We presented a solution to problem of path following for a mobile manipulator, namely GIM mobile machine, a fork lift. The proposed solution was so called time-coordinated path following and is compose of three subsystems visualized in Fig. 1. Subsystems are: joint trajectory generator, speed servo and path following, and coordinator. Given a desired path for the tool position and orientation, we presented a solution to generate corresponding joint trajectories. The solution is based on forward kinematics and simple integration of stable dynamics systems given by (1). We also described how we can prevent generating trajectories that cause collisions or exceed actuator limits. We then introduced concept of virtual time and presented path following and coordination laws. The solution is feedback based on forward kinematics and simple integration of stable dynamics systems given by (1). We also described how we can prevent generating trajectories that cause collisions or exceed actuator limits. We then introduced concept of virtual time and presented path following and coordination laws. We showed that such feedback control keep the end effector close to the desired path despite disturbances and faulty actuators. Future work includes tests on the real machine, which will require a systematic solution to tune the control gains since there are many control variables involved.

REFERENCES


