Transport Process Automation with Industrial Forklifts

Leopoldo Armesto Josep Tornero Juan Carlos Torres
Departamento de Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia
SPAIN

Abstract: This paper deals with transport process automation of industrial plants with industrial forklifts. In particular, a forklift has been completely automated based on external and internal sensors. A specific hardware architecture have been implemented and tested. Control software architecture includes the low level control algorithms decompounded by three control loops: position control of the steering wheel; speed control of the rear wheels and Cartesian position control of the whole vehicle. Finally, a teleoperation and a data monitoring application is described as a way of validating the implemented platform.

Key-Words: Transport Process, Industrial Forklift, Hardware and Control Architectures, Teleoperation.

1 Introduction

Industrial forklifts are used for management and transport processes automation in industrial plants. The two processes can be seen separately but with a significant degree of interaction between them. The management processes are based on database applications where client orders take place while the transport process consists on moving industrial forklifts according to the orders commanded by the management application.

There are several possible operating modes from manual with assistance to complete automation giving different degree of autonomy. In this sense, the main objectives of the transport process would be:

- Command the vehicles according to the movements established in the management application orders.
- Introduce high security levels defined as protection area around the vehicle.
- Implement obstacles collision detection and avoidance systems [2].

In a semi automated mode, the system have to:
- assist the driver in his movements and operation.
- help the driver in the interact with the management process application through the appropriate man-machine interface.

This paper is focused on the automation of an industrial forklift and is organized as follows. In section 2 we describe the sensors and actuators of the automated industrial forklift. The hardware and software is described in section 3. The control architecture of the vehicle is detailed on section 4. In section 5 a teleoperation and data monitoring application of the vehicle is showed. Conclusions are obtained in section 6.

The industrial forklift used is a FBT15 65-series vehicle from Nichiyu.

2 Description of Sensor and Actuators

The industrial forklift has been completely automated by introducing additional sensing and a controller system based on a PLC.

Originally, the vehicle [1] is compound by three DC motors driving three wheels, two at the front and the other at the rear part of the vehicle. An accelerator pedal commands an electronic differential which governs the two front wheels to avoid sliding. A gear establishes vehicle’s sense (forward, neutral or
reverse sense). The steering wheel position is originally commanded by a potentiometer, inducing a movement on the rear wheel DC motor. When moving, the rear wheel centers the potentiometer through a mechanical link, closing an internal control loop.

Two incremental encoders, one for each front wheel; and an absolute encoder for the steering wheel position measurement are introduced as additional sensing.

A manual/automatic mode selector has been installed connected to a relay circuit board as shown in Fig. 1.

Furthermore, two laser ranges, PLS from SICK, have been installed, one at the front and the other at the rear part of the vehicle (see Fig. 2). These sensors provide two digital inputs which are activated when an intrusion of two configurable areas occurs (warning and protection areas). Laser sensors also provide range information of a 180° scan with 0.5° and 5cm precisions via RS232 with a compatible PC baud rate up to 38400 bauds, taking approx. 250ms time delay on data request.

A ring of 6 US and 8 IR sensors have been distributed along the vehicle in order to cover lateral zones as shown in Fig. 2.

IR GP2D02 sensors transmit distance information up to 70cm, where the physical principle of this sensor is the inclination of the IR light reflected on objects. Furthermore, an study made with different colored objects demonstrated that in the worst case the precision was of ±1cm, even with natural light.

US sensors from Polaroid 600-Series are used to detect object presence up to 8m with an open angle of 30°. The ultrasonic wave is transmitted and received by the same transducer, therefore it is required a waiting time interval between the end of the transmission to the beginning of the reception in order to avoid false echoes. An amplified 50KHz burst pulse signal generated by a μController is send to the transducer. The received signal is amplified by the sonar receptor TL852. The power of the ultrasonic wave weaken for large distances, nevertheless, TL852 is able to keep sensibility for all the operating range. Distance measurement is based on the principle of time-on-the-fly (TOF).

US sensors are affected by the cross talking phenomenon, that is, an ultrasonic sensor receive a wave generated by other transducer. In order to avoid this effect, US sensor shots are made by pairs of opposite located sensors. According to Fig. 2, different sequences of shots can be established, i.e.: US1-US4, US2-US5, US3-US6 or IS1-US6, US2-US5, US3-US4.

Each IR and US sensor is integrated in a CAN node compounded by a PIC16F877 μController with MCP2510 and 82C250 as CAN controller and driver respectively. The CAN controller is connected to the μController via the SPI bus. All the CAN nodes are connected through a CAN bus ring communicating at 500Kbauds.

![Fig. 1. Relay circuit board](image1)

![Fig. 2. Ranger sensors location](image2)

### 3 Hardware Architecture
In this industrial vehicle, real-time and robust control requirements are implemented by closing the control loop with an industrial PLC, managing with sensors and actuators. In a higher level an industrial computer communicates with the PLC handling
a laser ranger via RS232 and IR and US sensors through the CAN field bus, as shown in Fig. 3.
A client application running on an on-board computer allows to communicate (wireless) with a server application for data monitoring.
We want to remark that the main criteria considered for the selection of this architecture has been the robustness introduced by the PLC. On the other side in the industrial computer, object detection and avoidance algorithms [2] and data fusion for map building applications [4] are easily implemented. The high hardware architecture cost using the PLC is completely justified on an industrial environment, where security and control are fundamental requirements.

4 Control Architecture

Control architecture is based on three control loops: rear wheel position control; rear wheel speed control and vehicle’s Cartesian position control.
The rear wheel position control loop includes the rear DC motor; the absolute encoder; a Kalman filter to estimate rotation speed of the wheel and a PID position controller.
The rear speed control loop is formed by the electronic differential which manage with the two DC motors (left and right), two incremental encoders (one for each wheel) and a PI speed controller.
References to these control loops are fixed by the position vehicle control loop, which generates appropriates the Cartesian velocities. These velocities are translated into the references of the other control loops using the inverse kinematic model.
In order to close position vehicle control loop, it is required a position estimation of the vehicle. This estimation is implemented by a Kalman filter using the direct kinematics equations.
In Fig 4, the described control architecture is shown, where $\overline{v}_{C_{1}}$, $\overline{v}_{C_{1}}$, and $\overline{v}_{C_{1}}$ are speed measurements of left and right wheels; $\beta$ is the rear wheel position; $\overline{v}_{C_{1}}$, and $\overline{v}_{C_{1}}$ are translational and rotational speeds of the vehicle; and $x,\dot{x},y,\dot{y},\dot{\theta}$ and $\dot{\omega}$ define the posture of the vehicle with respect to a Cartesian coordinates system.
In next subsections, only the two low-level control loops (rear wheel position and velocity) are described as well as vehicle kinematics and position estimation.
In the implementation of the third control loop a multi-rate schema [5] will be considered as control architecture. This is justified for the use of sensors with different sampling periods.
4.1 Vehicle kinematics

In this subsection, direct and inverse kinematic models of the vehicle are studied. The direct kinematic model expresses Cartesian vehicle velocities based on wheels speed using encoder information. The model of the industrial forklift is the well-known tricycle model [3] as shown in Fig. 5.

Therefore, we have sensed the directional wheel angle \( \beta \) and the two front wheel speed \( \tau_{V_{C_y}} \) and \( \tau_{V_{C_y}} \), obtaining redundant information for the direct kinematic model. This redundant information minimizes the residual error of inaccuracy measurements. The following equations have been used in order to compute \( \tau_{V_{C_y}} \), \( \tau_{V_{C_y}} \), and \( \tau_{V_{C_y}} \), using the encoders information:

\[
\tau_{V_{C_y}} = \frac{1}{X} \begin{cases} 
\frac{1}{2} |v_{y} + \frac{M}{2} \cdot \sin(2 \cdot \beta)| \tau_{V_{C_y}} + \left( \begin{array}{c} 3 \cdot L - \frac{M}{2} \cdot \sin(2 \cdot \beta) \\ -3 \cdot L - \frac{M}{2} \cdot \sin(2 \cdot \beta) \end{array} \right) \tau_{V_{C_y}} \\
\frac{1}{2} |v_{y} + \frac{M}{2} \cdot \sin(2 \cdot \beta)| \tau_{V_{C_y}} + \left( \begin{array}{c} 3 \cdot L - \frac{M}{2} \cdot \sin(2 \cdot \beta) \\ -3 \cdot L - \frac{M}{2} \cdot \sin(2 \cdot \beta) \end{array} \right) \tau_{V_{C_y}} 
\end{cases}
\]

(1)

\[
\tau_{V_{C_y}} = \frac{1}{X} \begin{cases} 
\frac{1}{2} \left( \begin{array}{c} (M' + 3 \cdot L') \cdot \cos(\beta) + M \cdot L \cdot \sin(\beta) \\ 2 \cdot L' \cdot \sin(\beta) \end{array} \right) \tau_{V_{C_y}} \\
\frac{1}{2} \left( \begin{array}{c} (M' + 3 \cdot L') \cdot \cos(\beta) - M \cdot L \cdot \sin(\beta) \\ 2 \cdot L' \cdot \sin(\beta) \end{array} \right) \tau_{V_{C_y}} 
\end{cases}
\]

(2)

where \( X = 6 \cdot L^2 + M^2 \cdot (1 + \cos(2 \cdot \beta)) \)

It is interesting to remark that, \( \tau_{V_{C_y}} \) is used to close the control loop, meanwhile \( \tau_{V_{C_y}} \) and \( \tau_{\omega_{R}} \) are used for the position estimation.

The inverse kinematic model generates the directional angle and the rear wheel speed references, \( \beta_{Re} \) and \( \tau_{V_{C_y}} \), for a given Cartesian speed vehicle, \( x_{\text{Ref}} \) and \( y_{\text{Ref}} \). The equations used are:

\[
\tau_{V_{C_y}} = \frac{1}{X} \begin{cases} 
\frac{1}{2} \left( \begin{array}{c} (M' + 3 \cdot L') \cdot \cos(\beta) + M \cdot L \cdot \sin(\beta) \\ 2 \cdot L' \cdot \sin(\beta) \end{array} \right) \tau_{V_{C_y}} \\
\frac{1}{2} \left( \begin{array}{c} (M' + 3 \cdot L') \cdot \cos(\beta) - M \cdot L \cdot \sin(\beta) \\ 2 \cdot L' \cdot \sin(\beta) \end{array} \right) \tau_{V_{C_y}} 
\end{cases}
\]

\[
\beta_{Re} = \arctan \left( \frac{M}{e} \cdot \frac{\cos(\theta - x_{\text{Ref}} \cdot \sin(\theta))}{\cos(\theta + x_{\text{Ref}} \cdot \sin(\theta))} \right)
\]

(3)

where \( e \) is the distance to a persecution point \( P \) located at the front of the vehicle used for position control as shown in Fig. 5.

4.2 Rear wheel position control

A discrete-time third order model for the rear DC motor driving the rear wheel has been identified at a sampling period of 50ms [5], being the input the signal applied to the DC motor \( u_k \) and the output the position measured with the absolute encoder \( y_k \):
\[ \int \beta_{1,1} = 1 \begin{bmatrix} 0.05 & 0 \\ 0 & 0.9592 \end{bmatrix} \int \beta_{1,1} + [0 \quad 0.069] u_k \] (4)

\[ y_k = [0 \quad 1] \begin{bmatrix} \int \tau y_{C,y} \\ \int \tau y_{C,y} \end{bmatrix} \] (7)

As it can be seen, the model contemplates the wheel position, its integrate and derivate variables. Based on this model an LQR controller has been obtained obtaining the following state feedback matrix:

\[ K = \begin{bmatrix} 2.1183 & 1.5315 \end{bmatrix} \] (8)

where \( \int \tau y_{C,y} \) is estimated with a simple discrete-time integrator.

Experimental results, shown in Fig 7, validates the designed control as shown in Fig. 6.

4.3 Rear wheel speed control

A discrete-time second order model with sampling period of 50ms has been identified using as input the acceleration signal \( u_k \) and as output the measure of the rear wheel speed \( y_k \):

\[ \begin{bmatrix} \int \tau y_{C,y} \\ \int \tau y_{C,y} \end{bmatrix} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.39484 \end{bmatrix} \begin{bmatrix} \int \tau y_{C,y} \\ \int \tau y_{C,y} \end{bmatrix} + [0 \quad -0.758] u_k \] (4)

\[ y_k = [0 \quad 1] \begin{bmatrix} \int \tau y_{C,y} \\ \int \tau y_{C,y} \end{bmatrix} \] (7)

In this case, another LQR regulator has been designed with the following state feedback matrix:

\[ K = \begin{bmatrix} 2.1183 & 1.5315 \end{bmatrix} \] (8)

Experimental results, shown in Fig 7, validates the designed control as shown in Fig. 6.

4.4 Position Estimation

A Kalman filter has been designed in order to estimate the vehicle position. In this case, a second order model of the vehicle is used:

\[ \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T^2/2 \\ 0 & 0 & 0 & 0 & T \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \] (9)

We are able to measure the following variables due to the direct kinematics model:

\[ \begin{bmatrix} x_m \\ y_m \\ \omega_m \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} [u_x, u_y, u_u] \end{bmatrix}_k \] (10)

We are able to measure the following variables due to the direct kinematics model:

\[ \begin{bmatrix} x_m \\ y_m \\ \omega_m \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} [u_x, u_y, u_u] \end{bmatrix}_k \] (10)

As it can be seen, the model contemplates the wheel position, its integrate and derivate variables. Based on this model an LQR controller has been obtained obtaining the following state feedback matrix:
respectively with covariance matrices \( Q = I \) and \( R = 0.5 \cdot I \).

We have applied the well-known Kalman filter recursive equations, as shown below,

\[
\begin{align*}
P(k+1/k) &= A \cdot P(k/k) \cdot A^T + B \cdot Q \cdot B^T \\
x(k+1/k) &= A \cdot x(k/k) \\
S(k+1) &= C \cdot P(k+1/k) \cdot C^T + R \\
K(k) &= P(k+1/k) \cdot C^T \cdot S^{-1}(k+1) \\
z(k) &= y(k) - C \cdot x(k+1/k) \\
x(k+1) &= x(k+1/k) + K(k) \cdot z(k)
\end{align*}
\]

(11)

The main advantage of using the Kalman filter is that in further implementation, laser scans measurements can be easily included in order to improve position estimation.

5 Teleoperation and Data Monitoring Application

A simple application has been created for teleoperating the industrial forklift and mainly for data monitoring. This is just a test application and not a final one running on the vehicle. This application has been implemented in Labview 6.1i and is located on the on-board computer. The control panel created can be transferred to another computer just by indicating the IP address of the wireless card.

In Fig. 8, it is showed the application control panel, where the trace of the vehicle is represented through a plot, and the rest of variables are represented with an appropriate indicator. As it can be seen an example of a whole round of 15x60m has been monitored with just 1.66m of error on Y axis.

6 Conclusions

This paper has described a completely automated industrial forklift based on external and internal sensors.

A specific hardware and software architecture have been implemented and tested using a teleoperation and a data monitoring application.

The paper has focused attention on the low level control algorithms: position control of the steering wheel; speed control of the rear wheels and Cartesian position control of the whole vehicle.

A Kalman filter for position estimation has been also implemented and validated.

Acknowledgment

We thank to L. Gracia, A. Herráez and S. Fuster for assisting in kinematic model, Labview interface and CAN implementation respectively.

References:


