A path following algorithm for mobile robots

**Tijmen Bakker · Kees van Asselt · Jan Bontsema ·
Joachim Müller · Gerrit van Straten**

Abstract This paper considers path following control for
a robotic platform. The vehicle used for the experiments
is a specially designed robotic platform for performing au-
tonomous weed control. The platform is four-wheel steered
and four-wheel driven. A diesel engine powers the wheels
via a hydraulic transmission. The robot uses a Real Time
Kinematic Differential Global Positioning System to deter-
mine both position and orientation relative to the path. The
deviation of the robot to the desired path is supplied to two
high level controllers minimizing the orthogonal distance
and orientation to the path. Wheel angle setpoints are de-
termined from inversion of the kinematic model. At low
level each wheel angle is controlled by a proportional con-
troller combined with a Smith predictor. Results show the
controller performance following different paths shapes in-
cluding a step, a ramp, and a typical headland path. A refined
tuning method calculates controller settings that let the robot
drive as much as possible along the same path to its setpoint,
but also limit the gains at higher speeds to prevent the closed
loop system to become unstable due to the time delay in the
system. Mean, minimum and maximum orthogonal distance
errors while following a straight path on a paving at a speed
of 0.5 m/s are 0.0, −2.4 and 3.0 cm respectively and the
standard deviation is 1.2 cm. The control method for four
wheel steered vehicles presented in this paper has the unique
feature that it enables control of a user definable position rel-
ative to the robot frame and can deal with limitations on the
wheel angles. The method is very well practical applicable
for a manufacturer: all parameters needed are known by the
manufacturer or can be determined easily, user settings have
an easy interpretation and the only complex part can be sup-
plied as a generic software module.

T. Bakker (✉) · K. van Asselt · G. van Straten
Systems and Control Group, Wageningen University,
P.O. Box 17, 6700 AA Wageningen, The Netherlands
e-mail: tijmen.bakker@tyker.com

Present address:
T. Bakker
Tyker Technology, P.O. Box 507, 6700 AM Wageningen,
The Netherlands

J. Bontsema
Wageningen UR Greenhouse Horticulture, P.O. Box 644,
6700 AP Wageningen, The Netherlands

J. Müller
Farm Technology Group, Wageningen University, P.O. Box 17,
6700 AA Wageningen, The Netherlands

Present address:
J. Müller
Institute for Agricultural Engineering, University of Hohenheim,
70593 Stuttgart, Germany

1 Introduction

In organic farming there is a need for weeding robots that
can replace manual weeding. The required labour for hand
weeding is expensive and often difficult to obtain. In 1998
in the Netherlands on average 73 hours of hand weeding
were spent on one hectare of sugar beet in organic farming
(Van der Weide et al. 2002). In this paper a path following
control system for a weeding robot is presented enabling the
robot to navigate autonomously along a path.

A common design for a control system for agricultural
vehicles is to split up the control system in a low level and a
high level controller (Bendtsen et al. 2002; Bak and Jakob-
sen 2004). The low level electro-hydraulic system is a sys-
tem with dead time. A well known method to compensate for

time delays is the Smith predictor (Stephanopoulos 1984). Ge and Ayers (1991) applied this successfully to control an electro-hydraulic system on a hydraulic test bench. We used a Smith Predictor to compensate for time delays in the application of an electro-hydraulic steering system in practice.

The high level control system is partly inspired by work of Hague and Tillett (1996) and Bendtsen et al. (2002). Bendtsen et al. (2002) used a model for a four-wheel steered vehicle derived from Campion et al. (1996) and presented simulation studies applying feedback linearization as a control method. Hague and Tillett (1996) worked out a method for path following for a vehicle with two driven wheels and two free rolling wheels. For a simplified vehicle model they developed a controller. From the output of this controller follow the wheel speed setpoints by inversion of the kinematic vehicle model. In this paper this method is worked out for a four wheel steered robot, using the kinematic model derived from Campion et al. (1996) resulting in wheel angle and wheel speed setpoints for the low level control system. A refined tuning method of the high level controller, adapted from Skogestad (2003), lets the robot drive as much as possible along the same path to its setpoint independent from speed, but also limit the gains at higher speeds to prevent the closed loop system from becoming unstable at higher speeds because of the time delay.

2 Robotic platform

2.1 Platform

The vehicle used for the experiments is a specially designed robotic platform for performing autonomous weed control (Fig. 1). The design of the platform was described earlier by Bakker et al. (2008). The platform is four-wheel steered and four-wheel driven. There is no mechanical constraint on the maximum turning angle of a wheel around its vertical axis,

Fig. 1 Robot platform

but the wheel angles should be constrained to prevent twisting of the cables of the wheel speed sensors. Power is provided by a diesel engine that powers the wheels via an hydraulic transmission. The hydraulic transmission consists of a pump supplying oil to eight proportional valves, each connected to one fixed displacement hydraulic motor. Four hydraulic motors are used to drive the wheels, the other four to steer the wheels. Computer control of the valves is achieved using pulse width modulation via two micro-controllers connected to a CAN bus. The pump/valves combination is a 'load sensing' system: the pressure drop over the valves controls the displacement of the pump via an hydraulic load sensing connection and is limited to a small value, independent of load pressure. The platform is further equipped with a hitch that can be lifted hydraulically. A second hydraulic pump mounted in series with the first, supplies oil to two valves: one for lifting the hitch, one for control of auxiliary implements. Computer control of the valves is achieved also via a micro controller connected to the CAN bus.

2.2 Electronics

The weeding robot electronics consists of 9 embedded controllers connected by a CAN bus using the ISO 11783 protocol. In the inside of every wheel rim a cogwheel is mounted for wheel speed measurement. The two magneto resistive sensors per cogwheel are placed in such a way that the direction of rotation can be resolved. The rotation of the wheels is measured by these sensors with a resolution of 100 pulses per wheel revolution. The wheel angle of each wheel is measured by a Kverneland 180 degree sensor with an accuracy of one degree. Per wheel a micro controller is mounted transmitting wheel speed and wheel angle via the CAN bus. Two GPS antennas are used to measure both vehicle position and orientation. Both are connected to a Septentrio PolaRx2eH RTK-DGPS receiver with a specified position accuracy of 1–2 cm and a specified orientation accuracy of 0.3 degrees (1σ). The two GPS antennas are mounted on a metal plate to prevent multipath errors. A base station with a Septentrio PolaRx2e RTK-DGPS supplies the RTK-correction signals via a radio connection to the Septentrio PolaRx2eH receiver. The position of the base station itself can be configured by a correction supplied by the service of a company called 06-GPS via GPRS. One embedded controller running a real time operating system (National Instruments PXI system) also connected to the CAN bus controls the vehicle. The GPS receiver, and a radio modem are connected to the PXI via RS232. The radio modem interfaces the remote control used for manual control of the weeding robot. The manual control is used for guaranteeing safety during field trials and for transportation to and from the field. Different colored lamps of the signal tower can be operated via a micro-controller to indicate the current status of the robot platform. The platform is further equipped with sensors

measuring diesel level, hydraulic oil level, engine temperature and hitch height. The PXI system gathers wheel angles, wheel speeds, GPS data, remote control data and hitch height and controls the vehicle by sending messages to the three micro controllers connected to the hydraulic valves. A safety system consisting of four red emergency switches at the corners of the vehicle and a remote switch, controls the valves to neutral position on activation, overruling the computer control.

3 Path following structure

The vehicle control consists of two levels. At high level the wheel angle setpoints and wheel speed setpoints are determined in order to decrease the deviation from the path and the error in orientation. At low level, controllers are used to realize the wheel angles and wheel speeds determined by the high level control.

The deviation and the orientation error of the robot from a path are determined by a specially designed orthogonal projection on the path using the measured orientation and the GPS position. The orthogonal projection is designed to calculate the deviation and the orientation error relative to a line of positions $y(x)$.

4 Low level control

4.1 Wheel angle process model

The low level control realizes for each wheel the wheel angle and the wheel speed. The hydraulic valves used for steering the wheels of the weeding robot have a certain reaction time, resulting in a time delay of the steering. Furthermore, if a valve has a commanded open time of less than the dead time, a control does not have any effect. So the wheel angle process can be represented by:

$$\dot{\beta} = 0 \quad \text{for } t_{open} < t_{dead} \tag{1}$$

$$\dot{\beta} = K_p \cdot u(t - t_d) \quad \text{for } t_{open} > t_{dead} \tag{2}$$

and:

$$u(t) = -1995 \quad \text{if } U < 2500$$

$$u(t) = U - 4495 \quad \text{if } 2500 \le U \le 4000$$

$$u(t) = 0 \quad \text{if } 4000 < U < 6000$$

$$u(t) = U - 5405 \quad \text{if } 6000 \le U \le 7500$$

$$u(t) = 2095 \quad \text{if } U > 7500$$

where:

$\dot{\beta}$ is the wheel steering angle speed [°/s].
K_p is the gain of the process and equals 0.0712.

u is the control corrected for the dead band.
U is the control [$\%U_{DC} \cdot 100$].
U_{DC} is the power supply voltage and equals about 12 [V].
t_d is the delay of the system and equals 0.25 [s].
t_{open} is the time generated by a counter counting the time that the commanded control is in the active band (outside the dead band where $4000 < U < 6000$). It resets when the commanded control returns to the dead band.
t_{dead} is the dead zone of the system and equals 0.15 [s].

The value of t_{dead} was determined from tests in which the open time of a valve was varied, K_p, t_d and the values that relate u to U followed from step responses of the system.

4.2 Wheel angle control

To compensate for the time delay a P controller with Smith predictor is used for the wheel steering control (Stephanopoulos 1984).

The wheel angle control of the robot was tested by applying setpoint changes to one wheel while the robot was standing still on a flat concrete floor. From some first measurements it appeared that at large setpoint changes the variable pump controlled by the load sensing system could not react fast enough for the change in the flow required to maintain full pressure in the hydraulic system. Furthermore, if we imagine the robot driving over the field, the flow needed for steering will require only small changes in the flow already present for driving. So to simulate the presence of a continuous oil flow for driving during the wheel angle control test, one wheel was lifted from the floor and a constant control was put on the valve controlling its speed.

The average error of a series of 96 measurements on a wheel angle setpoint change of 10 degrees decreased within one second to zero plus or minus 2 degrees (see Fig. 2).
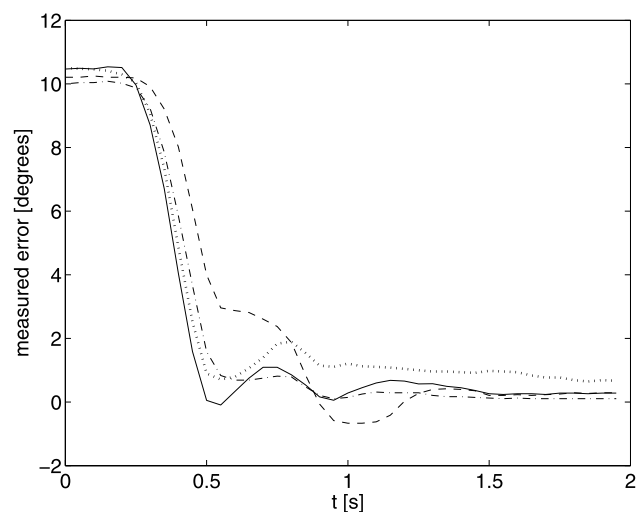

Fig. 2 Performance of the wheel angle control, average of 96 measurements. The setpoint changes at $t = 0$ s from 10 to 0 degrees (- - - front left , — rear left, · - · rear right, · · · front right)

5 High level control

5.1 Vehicle model

The point of the vehicle that should follow the path is the vehicle implement attached to the vehicle at a certain speed v. Consider a path-relative coordinate system (x_P, y_P) as illustrated in Fig. 3. The implement position is then completely described by $\xi = [x \ y \ \theta]^T$ where x denotes the distance along the path, y the perpendicular offset from the path, and θ the heading angle of the platform relative to the path (see Fig. 3).

Consider a coordinate system (x_v, y_v) fixed to the robot frame. The position of a wheel in this vehicle coordinate system is characterized by the angle γ_i and the distance l_i where i is the wheel index. The orientation of a wheel relative to x_v is denoted β_i. The model assumes pure rolling and non-slip conditions and driving in a horizontal plane. Therefore the motion of the robot can always be viewed as an instantaneous rotation around the instantaneous center of rotation (ICR). At each instant, the orientation of any wheel at any point of the robot frame must be orthogonal to the straight line joining its position and the ICR. The two-dimensional location of the ICR is specified by the angles of two wheels. For convenience a virtual front wheel β_f and a virtual rear wheel β_r is introduced with corresponding γ_f, l_f, γ_r and l_f, respectively located right in between the front wheels and right in between the rear wheels. The motion of the vehicle implement is described by the following state-space model
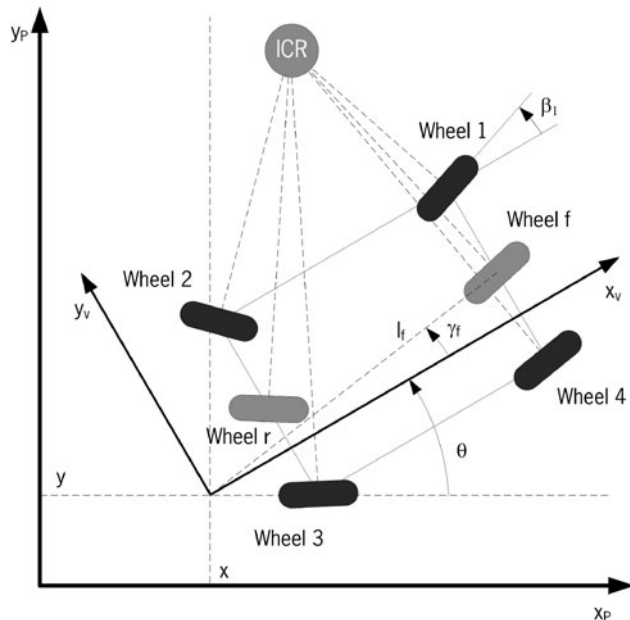

Fig. 3 Robot with ICR

derived from earlier work from Campion et al. (1996) and Bendtsen et al. (2002):

$$\dot{\xi} = R^T(\theta) \Sigma(\beta_i) \eta \tag{3}$$

where $R(\theta)$ is the orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

and:

$$\Sigma(\beta_i) = \begin{bmatrix} l_f \cos(\beta_r) \cos(\beta_f - \gamma_f) \\ \quad - l_r \cos(\beta_f) \cos(\beta_r - \gamma_r) \\ l_f \sin(\beta_r) \cos(\beta_f - \gamma_f) \\ \quad - l_r \sin(\beta_f) \cos(\beta_r - \gamma_r) \\ \sin(\beta_f - \beta_r) \end{bmatrix} \tag{5}$$

The scalar η is a velocity input. The wheel orientation β_3 and β_4 follow from β_f and β_r as described by Bendtsen et al. (2002) and Sørensen (2002) and β_1 and β_2 can be found in a similar way:

$$\beta_1 = \arctan\left(\frac{L \sin(\beta_f) \cos(\beta_r)}{L \cos(\beta_f) \cos(\beta_r) - \frac{1}{2} W \sin(\beta_f - \beta_r)} \right)$$

$$\beta_2 = \arctan\left(\frac{L \cos(\beta_f) \sin(\beta_r)}{L \cos(\beta_f) \cos(\beta_r) - \frac{1}{2} W \sin(\beta_f - \beta_r)} \right)$$

$$\beta_3 = \arctan\left(\frac{L \cos(\beta_f) \sin(\beta_r)}{L \cos(\beta_f) \cos(\beta_r) + \frac{1}{2} W \sin(\beta_f - \beta_r)} \right) \tag{6}$$

$$\beta_4 = \arctan\left(\frac{L \sin(\beta_f) \cos(\beta_r)}{L \cos(\beta_f) \cos(\beta_r) + \frac{1}{2} W \sin(\beta_f - \beta_r)} \right)$$

where L is the distance between the front and rear wheels and W the distance between the left and right wheels.

The wheel angular speeds $\dot{\phi} = [\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{\phi}_4]^T$ are controlled at low level, and follow from the vehicle model:

$$\dot{\phi} = J_2^{-1} J_1(\beta_i) \Sigma(\beta_i) \eta(t) \tag{7}$$

where:

$$J_1(\beta_i) = \begin{bmatrix} \cos(\beta_1) & \sin(\beta_1) & l_1 \sin(\beta_1 - \gamma_1) \\ \cos(\beta_2) & \sin(\beta_2) & l_2 \sin(\beta_2 - \gamma_2) \\ \cos(\beta_3) & \sin(\beta_3) & l_3 \sin(\beta_3 - \gamma_3) \\ \cos(\beta_4) & \sin(\beta_4) & l_4 \sin(\beta_4 - \gamma_4) \end{bmatrix}$$

$$J_2 = \begin{bmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 \\ 0 & 0 & r_3 & 0 \\ 0 & 0 & 0 & r_4 \end{bmatrix} \tag{8}$$

and r_1, r_2, r_3, r_4 are the radii of the four wheels.

According to Campion et al. (1996), η is an artificial variable without a physical meaning that can be interpreted as a velocity input. The unit of η is s^{-1}. However, in our system η has to be related to v. For this we have to bare in mind that v is defined by $\dot{x}$ and $\dot{y}$:

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \qquad (9)$$

If the robot is exactly following the path, $\dot{x} = v$. If the robot is moving perpendicular to the path $\dot{y} = v$.

From (3) and (9) then follows:

$$\eta = \frac{v}{\sqrt{([R^T(\theta)\Sigma(\beta_i)]_{11})^2 + [R^T(\theta)\Sigma(\beta_i)]_{21})^2}} \qquad (10)$$

For convenience let:

$$a = \sqrt{([R^T(\theta)\Sigma(\beta_i)]_{11})^2 + [R^T(\theta)\Sigma(\beta_i)]_{21})^2} \qquad (11)$$

and

$$\Sigma(\beta_i) = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \qquad (12)$$

From (3) and (12) then follows

$$
\begin{aligned}
R^T(\theta)\Sigma(\beta_i) &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_1 \cos(\theta) - \sigma_2 \sin(\theta) & 0 \\ \sigma_1 \sin(\theta) + \sigma_2 \cos(\theta) & 0 \\ \sigma_3 \end{bmatrix}
\end{aligned} \qquad (13)
$$

From (11) and (13) then follows:

$$a = \sqrt{\sigma_1^2 + \sigma_2^2} \qquad (14)$$

and from (10), (11) and (14):

$$\eta = \frac{v}{a} = \frac{v}{\sqrt{\sigma_1^2 + \sigma_2^2}} \qquad (15)$$

Substitution of (15) in (3) results in the following vehicle model:

$$\dot{\xi} = R^T(\theta)\Sigma(\beta_i) \frac{v}{\sqrt{\sigma_1^2 + \sigma_2^2}} \qquad (16)$$

5.2 Controller design approach

In our application the actuator determines the maximum robot speed possible. This is very common in agriculture operations: in some cases the soil and crop conditions limit the actuator velocity (e.g. when hoeing) and in other cases the actuator itself has properties limiting the maximum speed (e.g. mowers). So the robot should perform path following at a certain constant speed v. Therefore v is not regarded as a control variable, but as a parameter. Since v is fixed it follows from (9) only two differential equations remain, for y and θ.

The vehicle model given in (16) is a non-linear system which makes it hard to define and tune a simple controller. However, taking the inverse kinematic model, the non-linear system can be described by:

$$
\begin{aligned}
\dot{\theta} &= u_1 \\
\dot{y} &= u_2
\end{aligned} \qquad (17)
$$

by simply defining:

$$u_1 = \frac{v \sin(\beta_f - \beta_r)}{\sqrt{\sigma_1^2 + \sigma_2^2}} \qquad (18)$$

$$u_2 = \frac{v[R^T(\theta)\Sigma(\beta_i)]_{21}}{\sqrt{\sigma_1^2 + \sigma_2^2}} \qquad (19)$$

Although there is no mechanical constraint on the wheel angles we introduce the following constraint (see Fig. 4) to prevent twisting of the cables from the wheel speed sensors:

$$-0.5\pi < \beta_i < 0.5\pi \qquad (20)$$

The path following control consists of two controllers that calculate u_1 and u_2, and a method that inverts the kinematic model in (16) to calculate the low level setpoints $\beta_{i,sp}$. The inversion should result in one unique solution for the actual control that satisfies the constraints in (20).
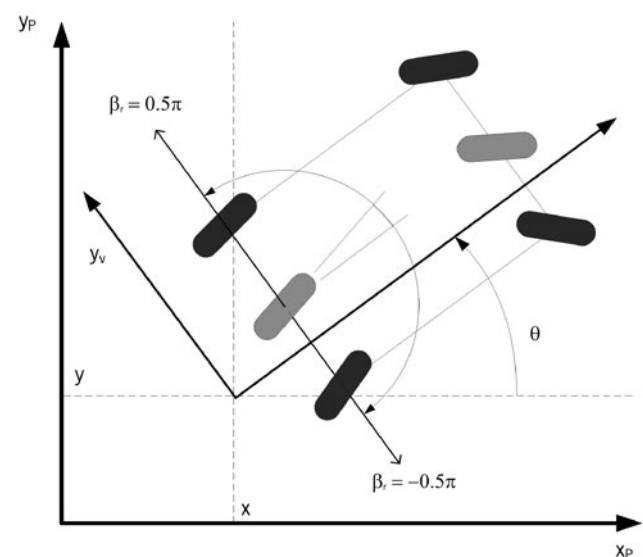

Fig. 4 Robot with wheel angle constraints: $-0.5\pi < \beta_i < 0.5\pi$

5.3 Model inversion algorithm

The low level setpoints β_f and β_r can be found from solving the system of the two non-linear equations (18) and (19), constrained by (20). Figures 5 and 6 show how u_1 and u_2 relate to β_f and β_r at $\theta = 0$ rad and $v = 1$ m/s.

The algorithm to solve β_f and β_r includes the following steps:

1. For every given β_f satisfying (20), where β_r is also satisfying (20) and $|\beta_f - \beta_r| \le 0.5\pi$, the allowed range for β_r is calculated by incrementing β_r with 0.01π and checking if $\beta_{1..4}$ calculated with (6) meet (20). Define $\beta^*_{r,max} = \mathrm{argmin}(\frac{u_1}{v}(\beta_r))$ and $\beta^*_{r,min} = \mathrm{argmax}(\frac{u_1}{v}(\beta_r))$, where $\frac{u_1}{v}$ is defined by (18), and calculate $\beta^*_{r,max}$ and $\beta^*_{r,min}$ for the allowed range of β_r. The calculation can be done independent of v, since $\frac{u_1}{v}$ is not depending on v. β_f is incremented with 0.01π resulting in a table with
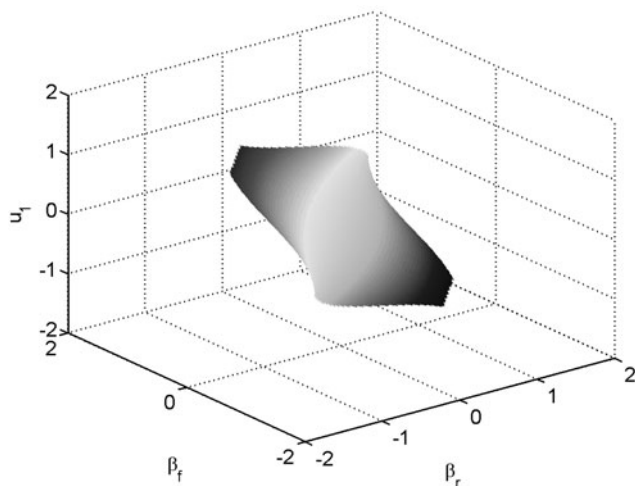

Fig. 5 Relation between β_f and β_r to u_1 at $\theta = 0$ rad and $v = 1$ m/s
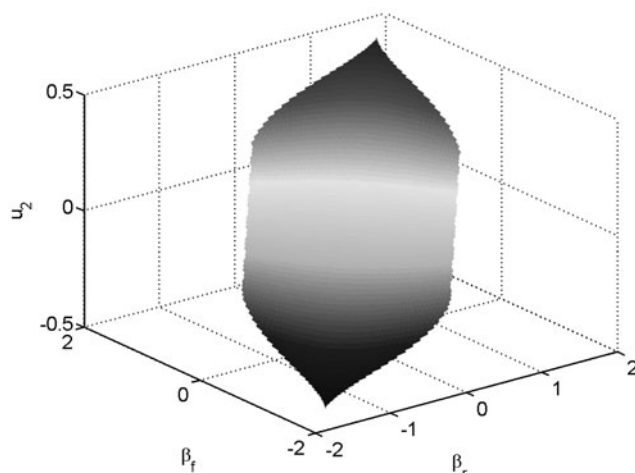
Fig. 6 Relation between β_f and β_r to u_2 at $\theta = 0$ rad and $v = 1$ m/s

for every β_f a $\beta^*_{r,min}$ and a $\beta^*_{r,max}$. The overall maximum and minimum feasible $u^*_{1,max,feas} = \max(\frac{u_1}{v})$ and $u^*_{1,min,feas} = \min(\frac{u_1}{v})$ is calculated by taking the maximum and minimum over the resulting table.

2. The robot has always to return to the path in the direction of the path even if the angle θ of the robot to the path is larger than 0.5π. If the measured angle θ of the robot to the path is larger than 0.5π it turned out that the robot will make a turn in the opposite direction to return to the path. To avoid this the θ used in the algorithm is constrained.

$$\theta = 0.5\pi \quad \text{for } \theta > 0.5\pi$$

$$\theta = -0.5\pi \quad \text{for } \theta < -0.5\pi$$

3. Constrain u_1 to feasible values:

$$u_1 = u^*_{1,max,feas} v \quad \text{for } \frac{u_1}{v} > u^*_{1,max,feas}$$

$$u_1 = u^*_{1,min,feas} v \quad \text{for } \frac{u_1}{v} < u^*_{1,min,feas}$$

4. Because u_1 is feasible, there exists at least one β_r that solves (18) for at least one β_f. For each β_f in the table developed under step 1, a β_r is searched that solves (18) by finding the zero of:

$$f_y(\beta_f) = u_1 - \frac{v \sin(\beta_f - \beta_r)}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{21}$$

where β_r is bounded by $\beta^*_{r,min}$ and $\beta^*_{r,max}$. A solution (β_f, β_r) is valid if $\beta_{1..4}$ calculated with (6) fulfills the constraint in (20) and if $|\beta_f - \beta_r| \le 0.5\pi$.

5. For this set of valid solutions (β_f, β_r) the corresponding u^*_2 is calculated with (19) where $\xi_1 > 0$. The solution with u^*_2 closest to the controller output u_2 is found by interpolation.

Step 1 has to be done only once for a robot configuration, e.g. at initialization of the system or the resulting table could even just be coded in the software. The reason is that step 1 is not depending on v. The results in the table are only dependent from the system parameters. Steps 2–4 are performed in realtime, because those depend from v and θ. The velocity v can be set by the user and θ is measured by the GPS, so these parameters change after robot startup.

The wheel speed setpoints $\dot{\phi}$ follow from (7), but are constrained by the minimum and maximum wheel speeds of the robot $\dot{\phi}_{min} = -6.9$ rad/s and $\dot{\phi}_{max} = 6.9$ rad/s. Therefore $\dot{\phi}$ is finally corrected for $\dot{\phi}_{min}$ and $\dot{\phi}_{max}$:

$$\dot{\phi}_i = \frac{\dot{\phi}_i}{\min(\dot{\phi}_i)} \dot{\phi}_{min} \quad \text{if } \min(\dot{\phi}_i) < \dot{\phi}_{min}$$

$$\dot{\phi}_i = \frac{\dot{\phi}_i}{\max(\dot{\phi}_i)} \dot{\phi}_{max} \quad \text{if } \max(\dot{\phi}_i) > \dot{\phi}_{max} \tag{22}$$

5.4 Controller design

The path tracking controller must thus compute u_1 and u_2 (and consequently $\beta_{f,sp}$ and $\beta_{r,sp}$) in order to minimize the orientation error to the path θ and the orthogonal distance to the path y.

Using (18) and (19) it follows that the dynamics for both θ and y are described by pure integrators. Therefore proportional (P) control for both θ and y was expected to be sufficient. Since we use a path-relative coordinate system, from a control point of view the setpoints are equal to zero. A controller is designed which computes u_1 and u_2:

$$u_1 = -K_\theta \theta \tag{23}$$

$$u_2 = -K_y y \tag{24}$$

In first instance for both K_y and K_θ constant values were used chosen by simulation to give the desired response and both were chosen equal to -1.

P control is not sufficient when there is a disturbance d on the input channel. The disturbance for a pure integrator with process gain K will result in a constant offset when using P control (with controller gain K_c):

$$y = \frac{K}{s + KK_c}d + \frac{KK_c}{s + KK_c}y_{sp} \tag{25}$$

and for a constant disturbance d:

$$y(t \to \infty) = \lim_{s \to 0} s \frac{K}{s + KK_c} \frac{d}{s} = \frac{d}{K_c} \tag{26}$$

So for reducing this offset, high K_c is needed, but a high K_c gives stability problems because of the delay in the loop.

The constant offset can be reduced by a PI controller, the transfer function is then:

$$y = \frac{Ks}{s^2 + KK_cs + KK_I}d + \frac{KK_cs + KK_I}{s^2 + KK_cs + KK_I}y_{sp} \tag{27}$$

and for a constant disturbance d:

$$y(t \to \infty) = \lim_{s \to 0} s \frac{Ks}{s^2 + KK_cs + KK_I} \frac{d}{s} = 0 \tag{28}$$

In simulation the influence of the controller gain K_c and the integration gain K_I was studied. For increasing K_I at constant K_c the overshoot resulting from a setpoint change increases, while the influence of the disturbance is reduced in a shorter time. However, at higher K_I, at a certain moment the transfer of the disturbance is overdamped, resulting in a worse overall response. If K_I is constant and K_c increases the overshoot is reduced. The K_c is further limited by the maximum controller output and the performance of the low-level control. Skogestad (2003) specifies the following PI settings for a pure integrator with time delay:

$$K_c = \frac{1}{K}\frac{1}{\tau_c + t_{d,hl}} \tag{29}$$

and:

$$\tau_I = 4(\tau_c + t_{d,hl}) \tag{30}$$

where K is the process gain, $t_{d,hl}$ is the time delay of the high level system, τ_I is the integral time of the controller and τ_c is the desired time constant of the controlled system. In our case the gains depend on speed because we want to let the robot drive independent of its speed as much as possible along the same path to its setpoint. The desired path can be viewed as a reference path or response curve in space, with a characteristic distance τ_c^* that plays the same role as a time constant in a response curve in time. Hence τ_c^* is the distance over which 63% of the ultimate response is realized. That means that:

$$\tau_c = \frac{\tau_c^*}{v} \tag{31}$$

So at low speeds mainly the speed v limits the controller gain K_c, while at higher speeds mainly the time delay $t_{d,hl}$ limits the controller gain K_c.

The manufacturer can choose to leave the setting of τ_c to the farmer, or give it a default value. For this default value he can use the suggestion of Skogestad (2003) of choosing $\tau_c = t_{d,hl}$. In our case the time delay in the high level control response $t_{d,hl}$ as determined from a hardware in the loop step response is 0.5 s (see Sect. 5.5). This can be explained from the sum of the low level control t_{dead} and t_d of 0.4 s (see Sect. 4.1) and another 0.1 s from both wheel angle measurement and control loop running at 50 ms interval at different nodes connected to the CAN bus. Choosing $\tau_c = t_{d,hl} = 0.5$ s at $v = 0.5$ m/s it follows then $K_c = -0.67$ and $\tau_I = 6.0$.

5.5 Evaluation methods

The performance of the path following control was tested in four different settings: Step responses with hardware in the loop simulation, step responses on pavement, a ramp on pavement and a typical headland path on pavement.

For hardware-in-the-loop tests the robot was mounted on trestles and the robot dynamics of the low level control were included in a closed loop. The robot motion in the x, y-plane was calculated by (16). A path with a step at 15 meters from the start was supplied. This distance was chosen to enable the robot to accelerate to its speed setpoint before arriving at the step. The robot was located at the start of the path with the same orientation as the path, was then set to autonomous control and followed the path autonomously. The orthogonal offset from the line was logged at a time interval of 50 ms.

The performance of the control was evaluated also with step responses on a paving. Four path coordinates that form a step were indicated on the pavement by chalk, measured out relative to a straight seam between the concrete plates

of the paving. Again the step was located about 15 meters from the start of the path to enable the robot to accelerate to its speed setpoint before arriving at the step. The GPS coordinates of the chalk crosses were then measured by locating a robot GPS antenna pole equipped with a plumb line above the chalk cross by manual control (see Fig. 1). The robot was then positioned on the step path consisting of the surveyed points by manual control, some meters before the step and approximately in the same direction as the path. The robot was then set to autonomous control and followed the path autonomously. Evaluation of the performance of path following was done in two ways: by logging the GPS coordinates, orthogonal distances and orientation errors during path following with a time interval of 50 ms and by measuring the real path drawn on the paving by a weeding unit with a chalk holder attached to the robot, holding about 2 cm wide chalks (see Fig. 7). To measure the chalk line, a rope was tightly stretched between two pins fixed in the seam between the concrete plates over the distance of the full path length and at regular distances of 25 cm along the rope the orthogonal distances from the rope to the middle of the chalk line were measured manually (see Fig. 7). At the step in the chalk line some more orthogonal distances were measured at distances of 5 and 10 cm from each other.

The performance of the control was evaluated further with a ramp response on a paving. Ramp following evaluates
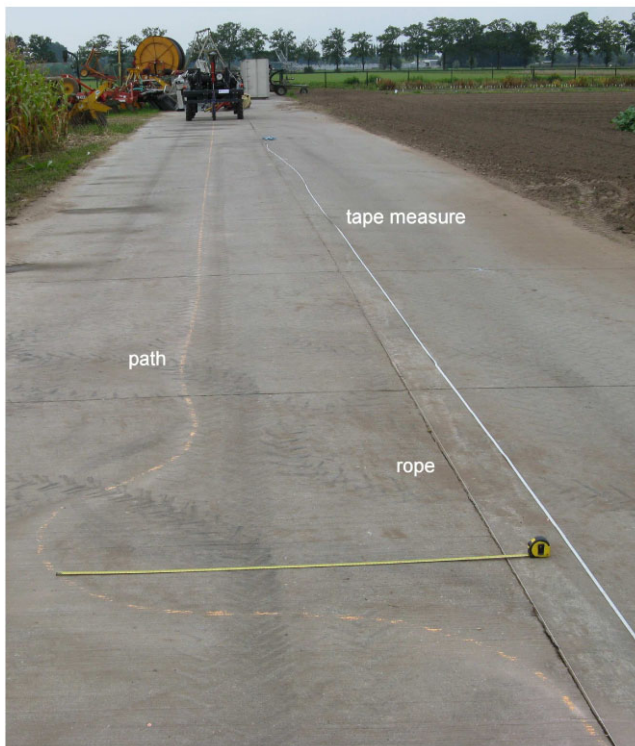
the response to orientation errors. The ramp was marked off as the diagonal of a square of 1 by 1 meter. Surveying the coordinates of the ramp was done in the same way as surveying the step. The robot position following the ramp is the actuator position at the back. Evaluation of the performance of ramp following was done by logging GPS coordinates, orthogonal distances and orientation errors during path following with a time interval of 50 ms.

Finally the performance of the control was evaluated with a typical headland path. The working width of the robot is 1.5 meters, so the paths that are connected by the headland path are only 1.5 meter apart, which results in a turn to the left resulting the left wheel entering the formerly left track before the turn again. The headland path is supplied as a sequence of 159 points, describing about half a circle with a radius of 0.75 meter. For step and ramp responses the position of the implement is following the path, but on the headland the robot middle position is following the headland path. The robot middle position is located exactly in between the four wheels. Evaluation of the performance of headland path following was done by logging GPS coordinates, orthogonal distances and orientation errors during path following with a time interval of 50 ms.

6 Experimental results

6.1 Hardware in the loop simulation with a P controller

The performance of the P controller with gain $K_c = -1$ was evaluated in an hardware-in-the-loop test. Results are visualized in Fig. 8. The controller lets the robot follow the path nicely. The mean orthogonal distance error between 8 and 15 m is 0.1 cm.
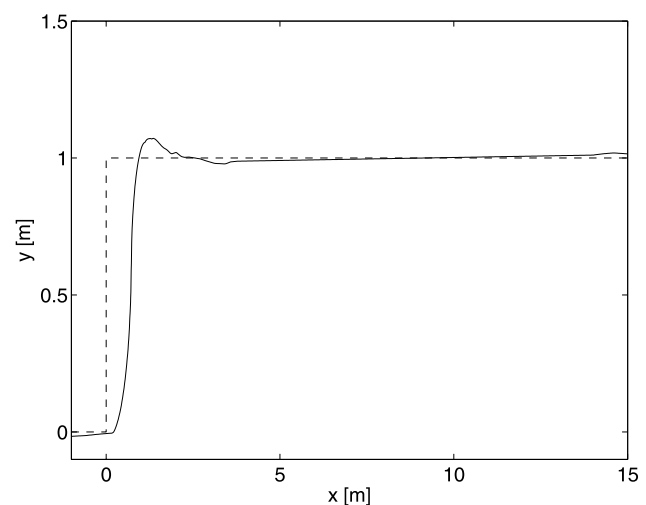

Fig. 7 Real path drawn on the paving by a weeding unit with a chalk holder attached to the robot. To the right of the robot the rope indicating the path is visible

Fig. 8 Hardware in the loop performance of path following with P control with $K_c = -1$ at $v = 0.5$ m/s (- - - step (desired path), — resulting path)

6.2 Step response on paving with a P controller

The performance of the P controller is tested on paving and both the GPS measurements and chalk line measurements are visualized in Fig. 9. Because of the step was marked off manually and there is a small error in the GPS measurement of the marked off corners of the step, the first and second part of the step are not exactly in parallel, so, for plotting the second part is aligned with the direction of the x-axis. The shape of the graph of the hand-measurement is very similar to the graph of the GPS-measurements: variations that are measured by hand are similar to variations that are measured by GPS. The offset between GPS and hand-made measurements at the start of the plot can be explained from the errors in surveying the marked-off corners of the step. The results in Fig. 9 show that there remains a static offset in the error that is not reduced by the controller: After the initial settling distance of $x = 8$ m, the mean error determined by GPS measurements over the distance 8 to 30 m is -1.6 cm and the minimum and maximum are -3.3 cm and 0.0 cm respectively and the standard deviation is 0.8 cm. The error in θ shows a very small static offset over this distance of $0.2°$,

see Fig. 10. The static offsets can be explained from a disturbance caused by the inaccuracy of the low-level wheel angle control. This inaccuracy of the wheel angle control can be explained from two main factors:

– Calibration errors of the zero position of the wheels. It appeared to be difficult to define the zero position of the wheels exactly.
– The accuracy of the control of the wheel angles is 2 degrees.

6.3 Step response with PI controller

Table 1 lists values for K_I and K_c corresponding with three values of τ_c^* at $v = 0.5$ m/s for which the performance of the PI controllers is tested. Figure 11 shows hardware in the loop step responses with these settings and Fig. 12 shows the step responses on pavement. In the test on paving it takes a longer distance before the system responds to the step than in the hardware in the loop test. This can be explained from an extra time delay of the position measurement in the paving test of maximal 0.2 seconds from the GPS receiver outputting data at 5 Hz. This extra time delay can also explain the
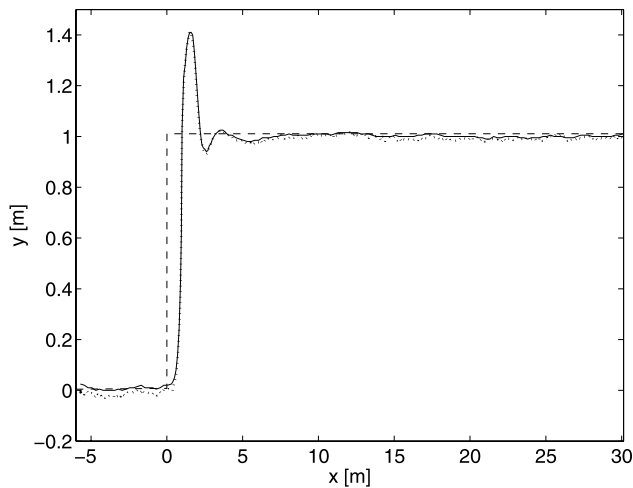

Fig. 9 Path following with P control with $K_c = -1$ at $v = 0.5$ m/s. The mean stationary offset between 8 and 30 m is 1.6 cm (- - - step, $\cdots$ resulting path measured by GPS, — resulting path by hand-measuring the chalk line)
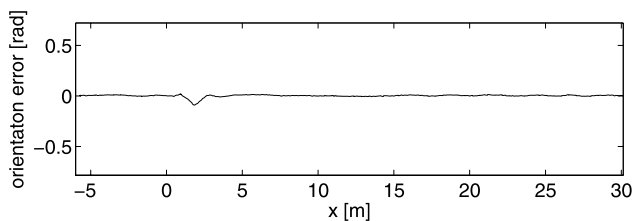

Fig. 10 Orientation error during path following with P control with $K_c = -1$ at $v = 0.5$ m/s. The mean offset between 8 and 30 m is 0.003 rad ($0.2°$)

Table 1 PI controller settings tested in a hardware in the loop setting

τ_c^*	0.3	0.4	0.5
τ_c	0.6	0.8	1
K_c	-0.91	-0.77	-0.67
τ_I	4.4	5.2	6.0
K_I	0.21	0.15	0.11
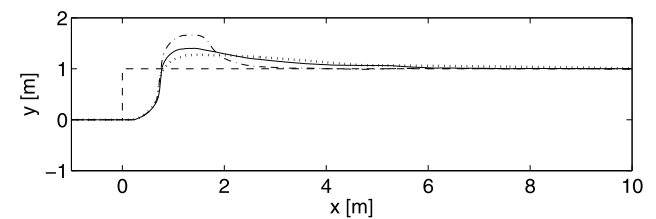
Fig. 11 Step response of y with PI control at $\tau_c^* = 0.3$ ($\cdot - \cdot$), $\tau_c^* = 0.4$ (—) and $\tau_c^* = 0.5$ ($\cdots$) obtained with hardware in the loop simulation at $v = 0.5$ m/s
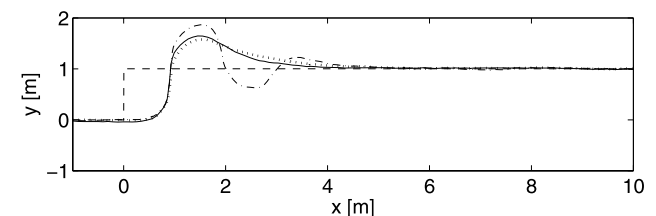
Fig. 12 Step response of y with PI control at $\tau_c^* = 0.3$ ($\cdot - \cdot$), $\tau_c^* = 0.4$ (—) and $\tau_c^* = 0.5$ ($\cdots$) obtained with tests on paving at $v = 0.5$ m/s
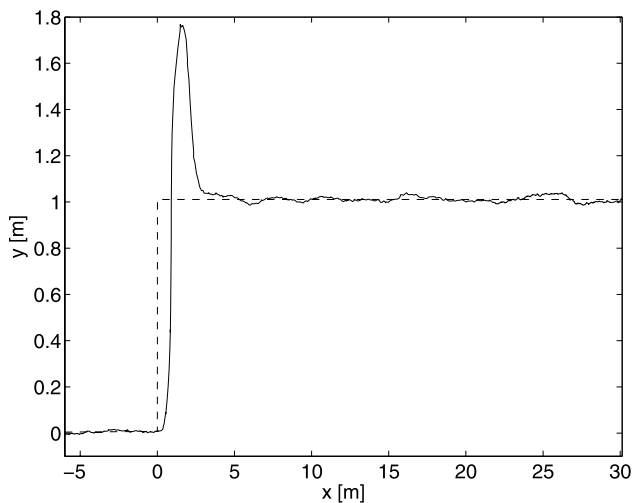
Fig. 13 Path following with PI control at $\tau_c^* = 0.4$ and $v = 0.5$ m/s. The mean stationary offset between 8 and 30 m is 0.0 cm
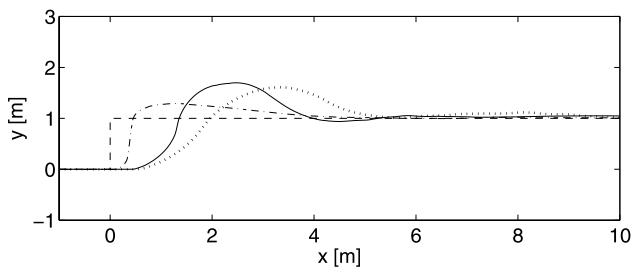

Fig. 14 Step response with the chosen controller setting $\tau_c^* = 0.4$ at $v = 0.25$ m/s ($\cdot - \cdot$), $v = 1.0$ m/s (—) and $v = 1.5$ m/s ($\cdots$) obtained with hardware in the loop simulation

Fig. 15 Step response with the chosen controller setting $\tau_c = 0.4$ at $v = 0.25$ m/s ($\cdot - \cdot$), $v = 1.0$ m/s (—) and $v = 1.5$ m/s ($\cdots$) obtained with tests on a paving

Fig. 16 Response at step of 0.1 m with the chosen controller setting $\tau_c^* = 0.4$ at $v = 0.25$ m/s ($\cdot - \cdot$), $v = 1.0$ m/s (—) and $v = 1.5$ m/s ($\cdots$) obtained with tests on a paving. Notice the scale

higher overshoot in the step responses on pavement compared to the hardware in the loop step responses. A possible solution for better simulating the real situation is to include this GPS time delay also in the hardware in the loop simulation.

To be able to compare the orthogonal distance error after the settling distance with the experiment with P control, the step on pavement was repeated with the finally chosen control setting $\tau_c^* = 0.4$, which means after 40 cm the robot is at 63% of its setpoint (see Fig. 13). The mean orthogonal distance error between 8 and 30 m is now 0.0 cm, the minimum and maximum are -2.4 cm and 3.0 cm respectively, and the standard deviation is 1.2 cm.

The controller settings were also tested at higher speeds: Fig. 14 shows results from hardware in the loop tests and Fig. 15 shows results from a test on paving. Again the effect of the extra time delay from of the GPS receiver in the paving test results in a longer reaction distance and a higher overshoot. A raise in speed from $v = 0.25$ m/s to 1.0 m/s increases the overshoot, but the overshoot does not further increase when the speed is further raised to 1.5 m/s. At higher speed equation (29) nicely reduces the τ_c but K_c and τ_c go
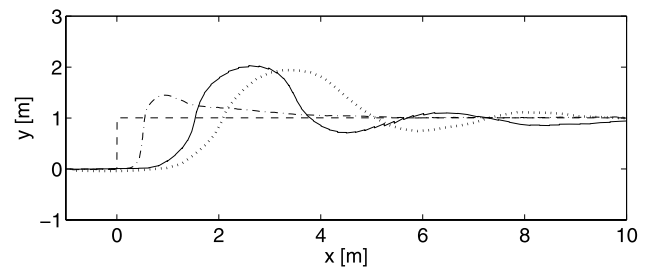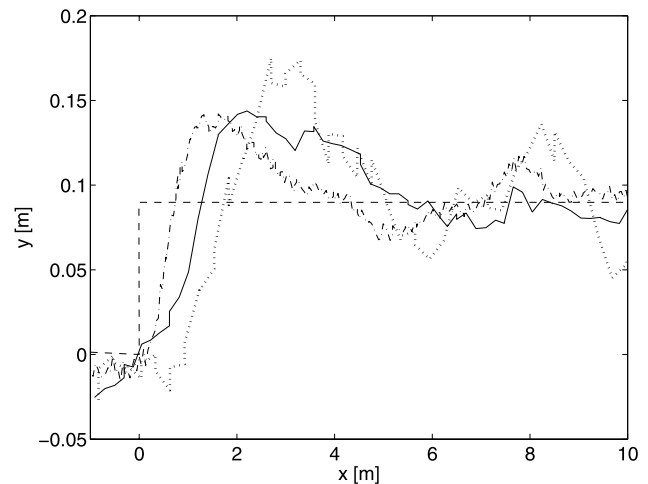
to fixed values resulting from $t_{d,hl}$, resulting in a stable response.

During path following on a field while performing weed control large step changes in the path will not occur and also large deviations from the path will not occur. Maximum error will only be around 0.1 m. Also under these circumstances the controller should perform well. Therefore a path following trial with a step of 0.1 meter was performed at several speeds on a paving. Results in Fig. 16 show besides the noise at this small scale including the noise of the GPS receiver a return to the path plus and minus 3 cm after a distance of 5 m for $v = 0.25$ m/s and $v = 1$ m/s which is satisfying.

6.4 Ramp response

A step in orientation was tested by performing a ramp response on a paving with PI controller settings $\tau_c^* = 0.4$ at $v = 0.2$ m/s (see Fig. 17). The maximum orthogonal error is 50 cm and the maximum orientation error is 1.13 rad.
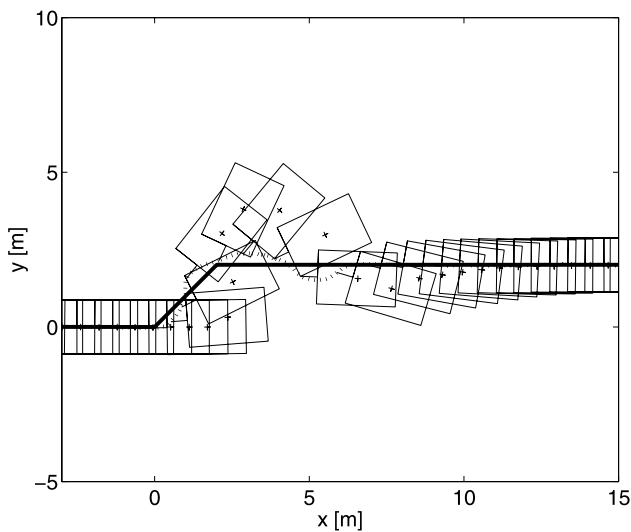
Fig. 17 Ramp setpoint (—) and resulting response (···) with the chosen controller setting $\tau_c = 0.4$ at $v = 0.2$ m/s obtained with a test on paving
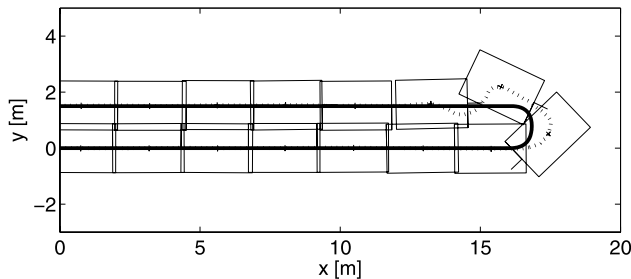

Fig. 18 Driven route (···) at headland path (—) following with P controller gain $K_c = -0.77$ at $v = 0.5$ m/s obtained with a test on paving

6.5 Headland path following

During turning the errors are increasing quite rapidly because of the sharp turn (turning radius of the path is 0.75 m) and the constraints on the wheel angles. To avoid overshoot due to the integrating action and because fast response during turning is much more important then high accuracy, P controllers are used. The possible offset at the end of the turn is then removed by the PI controller along the new straight path. The P controller gains for both the orthogonal distance controller and the orientation error controller are $K_c = -0.77$. Figure 18 shows the headland path, the driven route and the robot driving the path at $v = 0.5$ m/s plotted at a time interval of 3 seconds. Figures 19 and 20 show the orthogonal distance to the headland path and the orientation error while following the headland path. The maximum orthogonal distance and the maximum orientation error during the turn are respectively 69 cm and −0.93 rad. This headland path with such a limited radius can not be realized by the robot controller due to the constraints on the wheel angles.
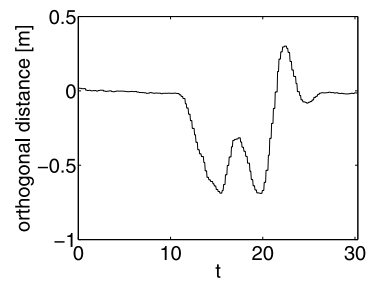

Fig. 19 Orthogonal distance error at headland path following with P controller gain $K_c = -0.77$ at $v = 0.5$ m/s obtained with a test on paving
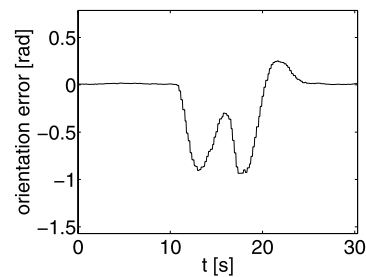

Fig. 20 Orientation error at headland path following with P controller gain $K_c = -0.77$ at $v = 0.5$ m/s obtained with a test on paving

However, on the headland it is less important to follow the headland path exactly than arriving in front of the next path again. After the sharp turn in the headland path, the errors are reduced very soon, the turning itself takes only about 15 seconds (from $t = 11$ s to $t = 26$ s). The orientation error and orthogonal distance error at the end of headland path following are 0.01 rad (1°) and 1.4 cm respectively.

7 Discussion

The difference in step responses of the hardware in the loop tests and tests on paving can be explained from the fact that the GPS time delay is not included in the hardware in the loop simulation.

The smaller time delay in hardware in the loop did also affect the chosen controller setting $\tau_c^* = 0.4$. While the performance is very good when following a straight line, the overshoot in the step responses is rather large: at normal working speeds for weeding from 0.25 m/s to 1 m/s and taking a small step of 10 cm the overshoot is about 5 cm. A higher τ_c^* chosen about equal to a total time delay including the GPS receivers' time delay should result in less overshoot and an even better result.

Although the path following performance is satisfying for the application of weeding, the results point out that if further improvements would be required for some other application these improvements should focus on reducing time

delays in the system. Ways to do this could be increasing the GPS receiver frequency, introduce Kalman filtering by using also wheel angle, wheel speed and gyro data, and by changing the flow compensated hydraulic valves to faster standard proportional valves.

Another improvement of path following accuracy would be to improve the low level wheel angle control. The limited steering accuracy of 2 degrees is due to the frequency of the electronic control units (20 Hz), hydraulic valve gain and the minimal required wheel angle controller output. The accuracy could be improved by changing the hydraulic valves to valves with a lower gain, but will be limited then again by the current angle sensor with an accuracy of one degree. So to obtain higher accuracies than one degree, both hydraulic valves and angle sensors should be changed.

The GPS-made measurements correspond very well with the hand-made measurements of the chalk line. Even small variations in the driven path are visible in both measurements. The largest error source causing the two measurements not to correspond exactly is the error in measuring the GPS coordinates of the corners of the step. The inaccuracy of a few centimeters in measuring these positions result from GPS accuracy, the GPS antenna height of about 2 meters above the paving surface, small not measured roll and pitch and the accuracy of about 1 cm by which the lead line connected to the GPS antenna pole could be positioned above the mark indicating the position to be measured. But it is very clear that the GPS measurements are perfectly usable for controller development, even handmade measurements are difficult to be made more precise than the RTK-DGPS ones.

To handle high errors that arise during headland path following due to robot constraints and path shape, P control was used to avoid overshoot due to the integrating action and because fast response during turning is much more important then high accuracy. To handle small errors while following a straight path with high accuracy, PI control was needed to overcome small errors from the low level control. This suggests a more generic approach that uses P control for large errors and PI control when the error is below a certain limit.

The high level control parameter τ_c^* can just be set by the user and controller tuning parameters follow then from the method adapted from Skogestad (2003). In contrary to a feedback linearization-based approach like Bendtsen et al. (2002), this approach gives the possibility to deal with limitations on the wheel-angles as they exist on almost every four-wheel steered vehicle in practice.

8 Conclusion

The Smith predictor compensates well for time-delays in electro-hydraulic steering systems in practice.

The control method for four wheel steered vehicles presented in this paper has a number of attractive features:

- Enables control of a user definable position relative to the robot frame which usually is the position of the implement.
- Can deal with limitations on the wheel angles.
- Has a good performance as shown in step responses made in a hardware in the loop fashion, on a paving and by following a typical headland path.

The method is a practical application for a manufacturer:

- The time delay can be determined from a step response at the factory, possibly to increase with the time delay of the GPS.
- The model inversion requires only some dimensions of the robot and its wheels. These are all parameters that are known by the manufacturer. The inversion of the vehicle model (the only complex part) could be supplied as a generic software module.
- The user supplies v and τ_c^*, where τ_c^* has an easy interpretable meaning: it is the driven distance along the path where 63% of the setpoint is reached. The method adapted from Skogestad (2003) determines then the controller settings and even suggests a default value for τ_c^*.

In contrast of the expected sufficiency of P control because of the dynamics of the processes for both θ and y act as pure integrators, PI control is needed to overcome small errors caused by inaccuracies of the low level wheel control.

The refined tuning method calculates controller settings that let the robot drive as much as possible along the same path to its setpoint, but also limit the gains at higher speeds to prevent the closed loop system to become unstable because of the time delay.

Mean, minimum and maximum orthogonal distance errors while following a straight path on a paving at a speed of 0.5 m/s are 0.0, −2.4 and 3.0 cm respectively and the standard deviation is 1.2 cm. Further improvements in accuracy of path following of a straight path are not to be expected, because of the standard deviation of 1.2 cm is about equal to the RTK-DGPS accuracy.

Additional research should show if the performance of the control will also be sufficient on a field with uneven soil surface. Combining the path following with automatic headland detection and automatic headland path generation should enable the robot to navigate autonomously over a whole field.

References

Bak, T., & Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering*, *87*(2), 125–136.

Bakker, T., Van Asselt, C. J., Bontsema, J., Müller, J., & Van Straten, G. (2008). Systematic design of an autonomous platform for robotic weeding. *Journal of Terramechanics*, *47*(2), 63–73. doi:10.1016/j.jterra.2009.06.002.

Bendtsen, J., Andersen, P., & Pedersen, S. (2002). Robust feedback linearization-based control design for a wheeled mobile robot. In *6th international symposium on advanced vehicle control (AVEC '02)*, Hiroshima, Japan.

Campion, G., Bastin, G., & D'Andra-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, *12*(1), 47–62.

Ge, J., & Ayers, P. D. (1991). Design of controllers for nonlinear electrohydraulic systems with time delays. *Transactions of the ASAE*, *34*(3), 1016–1022.

Hague, T., & Tillett, N. D. (1996). Navigation and control of an autonomous horticultural robot. *Mechatronics*, *6*(2), 165–180.

Skogestad, S. (2003). Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, *13*(4), 291–309.

Sørensen, M. (2002). Modelling and control of wheeled farming robot. M.Sc. thesis, Aalborg University, Aalborg.

Stephanopoulos, G. (1984). *Chemical process control. An introduction to theory and practice*. London: Prentice/Hall International.

Van der Weide, R., Lotz, L. A. P., Bleeker, P., & Groeneveld, R. M. W. (2002). Het spanningsveld tussen beheren en beheersen van onkruiden op biologische bedrijven. In F. Wijnands, J. Schroder, W. Sukkel, & R. Booij (Eds.), *Themaboek 303. Biologisch bedrijf onder de loep* (pp. 129–138). Wageningen: Wageningen Universiteit.

Jan Bontsema received a M.Sc. in Mathematics in 1982 and a Ph.D. in Mathematics and Physics in 1989, both from the University of Groningen. He has held positions at the Universities of Delft, Groningen and Wageningen. Since 1997 he works at Wageningen UR Greenhouse Horticulture, currently as senior researcher greenhouse climate control. His research interests include control engineering, robotics for agriculture and process monitoring in greenhouse horticulture.

Joachim Müller is Professor of Agricultural Engineering at Hohenheim University (Stuttgart, Germany) where he holds the Chair of Agricultural Engineering in the Tropics and Subtropics. He received his Ph.D. degree in 1992 in Renewable Energy Application and did research in Irrigation Technology as Assistant Professor at Hohenheim University until 2001. From 2001 through 2004 he held the Chair of Farm Technology at Wageningen University (The Netherlands) where he did research in Precision Framing and organized the first Field Robot Event.

Tijmen Bakker received a M.Sc. degree in Agricultural Engineering in 1999 from Wageningen University. After finishing his studies he worked at Kverneland Mechatronics BV in Nieuw-Vennep, the Netherlands. His work concerned the development of electronic control systems for agricultural machinery. In 2002 he started a Ph.D. research at the collaborating groups Systems and Control and Farm Technology of Wageningen University. The research concerned the design of an autonomous robot for weed control. He received a Ph.D. degree in 2009. In 2008 he founded the company Tyker Technology, by which he targets on research and development of technology for (semi) autonomous vehicles.

Gerrit van Straten (Dutch, 1946), holds an M.Sc. in Chemical Engineering from Eindhoven University, and a Ph.D. from Twente University, The Netherlands. He worked several years on environmental dynamics at Twente University and the International Institute for Applied Systems Analysis (IIASA) in Laxenburg (Austria). In 1990 he accepted a position as full professor at Wageningen University, where he leads the Systems and Control Group. The group develops and applies systems and control methodology to study the behaviour of dynamical systems in the bio- and agrosciences, and to realise automated systems in the agro-, environmental and food industry. He received the IFAC Outstanding Contribution Award, and served as Chair of the IFAC Technical Committee on Control in Agriculture (2003–2008). He is also Associated Editor for Control Engineering Practice (CEP) and Editor-in-Chief of Computers and Electronics in Agriculture (COMPAG).

Kees van Asselt received his B.Sc. degree in Electronics in 1976 and is working since then at the System and Control Group of Wageningen University. His lecture subjects are Electronics, Computer Architecture and Sensortechnology. His research topics are mechatronics, autonomous vehicles and weed control.