Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computers and Electronics in Agriculture 44 (2004) 223–237

Computers and electronics in agriculture

www.elsevier.com/locate/compag

On-line near minimum-time path planning and control of an industrial robot for picking fruits

L.G. Van Willigenburg[a,*], C.W.J. Hol[b], E.J. van Henten[c]

[a] *Systems and Control Group, Wageningen University, Technotron, P.O. Box 17, 6700 AA Wageningen, The Netherlands*
[b] *Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*
[c] *Department of Greenhouse Technology, Institute of Agricultural and Environmental Engineering (IMAG B.V.), P.O. Box 43, NL-6700 AA Wageningen, The Netherlands*

Received 19 March 2003; received in revised form 24 September 2003; accepted 22 May 2004

Abstract

To be competitive, an industrial robot picking fruits must be able to perform this task in an amount of time which compares to that needed by humans. Because the location of a fruit changes due to the picking of others, the determination of their location has to be performed on-line and also the associated path planning for the robot. This poses two major problems in the development of fruit-picking robots since locating fruits and path planning, in general, are computationally expensive operations. This paper contributes to relaxing the second problem. Using the fact that the motions of the links of an industrial robot are approximately decoupled, this paper proposes a new method for near minimum-time path planning and control in the presence of obstacles. This method is computationally cheap compared to methods that solve the more general problem. Experimental results are presented which indicate the feasibility of this approach to make the robot competitive.

Keywords: Fruit picking; Path planning; Robots; Minimum time

1. Introduction

Due to the increasing labor costs of human fruit pickers and the increasing difficulty in employing them, there is an increasing demand for automatic picking of fruits. Recently,

* Corresponding author. Tel.: +31-317-482941; fax: +31-317-484957.
E-mail address: gerard.vanwilligenburg@wur.nl (L.G. Van Willigenburg).

0168-1699/$ – see front matter
doi:10.1016/j.compag.2004.05.004

an industrial robot has been adapted and developed that successfully picks cucumbers autonomously, but at a speed which is about an order of magnitude smaller than that of a human (Van Henten et al., 2002, 2003a). The on-line computation time required for detecting cucumbers and the computation time required for on-line path planning exceed the actual motion time of the robot. So, to increase the speed of autonomous fruit picking, it is most important to speed up the detection, which is performed by a moving camera connected to a vision system, and to speed up the path planning. This paper contributes to the latter by computing collision-free near minimum-time robot motions using a more efficient kinematic path planner than the one reported by Van Henten et al. (2003b). Despite the fact that considerable research effort has been spent on automatic harvesting of vegetable fruits using multiple degrees-of-freedom manipulators (Kondo et al., 1996; Hayashi and Sakaue, 1996; Arima and Kondo, 1999), to our best knowledge, apart from Van Henten et al. (2003b) and Ryuh and Ryu (1995), collision-free motion planning has not received any attention within this research area. This also applies to the computation of minimum-time robot motions.

Industrial robots are very often still controlled by PID controllers which control each robot link individually, without considering the influence of the motions of the other links, which act as disturbances to these PID controllers. The reason for choosing this control approach probably is that it is classical, easy to understand, and easy to implement since it does not require any complicated modeling of the robot. As a result, it is computationally relatively cheap.

Of course, all this comes at a price. Industrial robots are designed such that the motions of the links do not interact much. To achieve this, high gear ratios between the motors and links are applied. As a result, the energy efficiency and the speed of industrial robots are very poor. This is for instance reflected by the fact that there is a very poor ratio between the weight a robot can carry and its own weight. Typically, a robot weighing something like 50 kg can only carry weights of 2 kg. The reason is not just the high gear ratios, but also the fact that with larger loads, the PID control becomes inaccurate or even unstable. To control robots which do not have high gear ratios, advanced control based on mechanical models of the robot is needed to deal properly with the interacting link motions. Although this type of robot control is very well developed (Meier and Bryson, 1987; Geering et al., 1986; Van Willigenburg, 1991, 1993; Hol et al., 2001; Bobrow et al., 1985; Shin and McKay, 1985, 1986; Zlajpah, 1996), it is still not very often implemented in practice. Probably, these advantages do not sufficiently outweigh the increased complexity of the control in most robot applications. This for instance holds true in the case of fruit picking, where, at present, the speed of operation is not limited by the mechanical properties and control of the robot, but by the on-line computation time required for the detection of cucumbers and path planning. Because fruit-picking robots have, like most industrial robots, high gear ratios and are controlled by separate PID controllers for each link, the motions of the robot links can be assumed to be completely decoupled.

In this paper, we develop a new method for near time-optimal path planning amongst obstacles under the assumption that the robot links are decoupled. The difference between our approach and many others that have been presented in the literature (Meier and Bryson, 1987; Geering et al., 1986; Van Willigenburg, 1991, 1993; Hol et al., 2001; Bobrow et al., 1985; Shin and McKay, 1985, 1986; Zlajpah, 1996) is that our approach is much less sophisticated but computationally much cheaper. Most methods that have been proposed either consider time-optimal motions without obstacle avoidance (Meier and Bryson, 1987; Geering et al., 1986; Van Willigenburg, 1991; Hol et al., 2001) or they consider time-optimal motion along a pre-specified path (Van Willigenburg, 1993; Bobrow et al., 1985; Shin and McKay, 1985, 1986; Zlajpah, 1996). The latter, in general, does not result in a time-optimal solution because the selection of a time-optimal path interacts with the time-optimal control along this path. Our approach falls in this latter category. However, if a path amongst the obstacles exists which is a straight line in configuration space, our method is truly time-optimal. Note that true time-optimal control amongst obstacles is an optimal control problem with state constraints, which is in general very difficult to solve.

Most time-optimal control approaches are only tested in simulation (Meier and Bryson, 1987; Geering et al., 1986; Van Willigenburg, 1991; Hol et al., 2001; Bobrow et al., 1985; Shin and McKay, 1985, 1986; Zlajpah, 1996) and often on robots with no more than two degrees of freedom (Meier and Bryson, 1987; Geering et al., 1986; Van Willigenburg, 1991, 1993). In this paper, both simulation and experimental results are presented obtained with an industrial robot with five degrees of freedom. These experiments show the feasibility of our approach for on-line path planning and control as required by fruit picking.

Path planning of robots is usually performed in the so-called configuration space of the robot, which roughly means in the coordinates associated to each robot link rotation or translation. The basic purpose of this paper is to show the following: if the motions of the robot links do not at all interact a minimum-time motion from one point in the configuration to another is along a straight line in the configuration space. If the robot links do interact minimum-time motions, in general, are not straight lines in the configuration space. The determination of straight lines in the configuration space, which avoid obstacles, is performed by so-called kinematic path planners, some of which are very efficient computationally. In this paper, we propose to use a computationally very efficient kinematic path planner called Ariadne's Clew Algorithm (Bessiere et al., 1995; Mazer et al., 1998) and next, based on decoupled dynamic models of the individual robot links, we are able to compute the control which brings the robot from the initial to the final point along these lines in minimum time. If there is no straight line that avoids obstacles in between the initial and final position in configuration space, then a sequence of straight lines is used. This sequence of lines is also generated by the kinematic planner. A serious disadvantage of using a sequence of straight lines is that the robot has to stop at each point connecting one line to the next. Due to the intermediate stops, our computational approach does not compute true minimum-time motions. This is the price we have to pay to be able to use computationally very efficient kinematic planners that perform collision checks by linear interpolation.

The objective of this paper is not to make a comparison between all kinds of collision-free (kinematic and/or time-optimal) path planners, but just to show that our technique enables a very fast computation of a near time-optimal control, once we have a collision-free path consisting of straight lines in the configuration space. Fortunately, such paths are obtained from kinematic path planners which can be very efficient computationally, compared to other path planning algorithms (Bessiere et al., 1995, Mazer et al., 1998).

The outline of the paper is as follows. In Section 2.1, we present the dynamic models of the individual links of the robot and also two types of associated electrical and mechanical constraints. In Sections 2.2 and 2.3, the control that drives the robot in minimum time from one to another location along a straight line in the configuration space is computed. In Section 2.2, only one type of constraint is considered and it is shown that the need to move along a straight line in configuration space does not restrict the minimum motion time. In Section 2.3, both types of constraints are considered and it is argued that only in exceptional cases, the need to move along a straight line in configuration space does restrict the minimum motion time. In case of a sequence of lines, since the robot has to stop at each interconnection, the minimum time is simply the sum of the minimum times needed to travel the individual lines. Section 3 describes the implementation of this algorithm on a five degrees of freedom industrial robot (the Eshed Robotec MK2) and presents experimental results related to fruit picking and the associated computation times required by our computational procedure. Conclusions and suggestions for future research are presented in Section 4.

2. Materials and methods

2.1. Dynamic modeling of robots with approximately decoupled link motions

The very high gear ratios applied in most industrial robots causes the inertia of the gears and motors to dominate those of the links. As a result, the motions of the links of the robot hardly influence one another. The individual PID controllers for each link further suppress the interaction and non-linear behavior (Craig, 1986). Therefore, the following set of linear uncoupled models may be used to describe the dynamic behavior of an industrial robot with m links,

$$M_i \ddot{\Theta}_i = \tau_i, \quad i = 1, 2, \ldots, m \tag{1}$$

Depending on the type of link in Eq. (1), Θ_i represents the translation or rotation of the i^{th} link, M_i represents its effective inertia, and τ_i the motor torque. In general, the angles are restricted to certain intervals

$$\Theta_i \in \left[\Theta_i^{\min}, \Theta_i^{\max}\right], \tag{2}$$

where $\Theta_i^{\min}$, $\Theta_i^{\max}$ represent the lower and upper bounds, respectively. As argued by Van Willigenburg (1993), in the case of electrical motors, it is realistic to consider the motor torques as control variables since these are approximately proportional to the motor currents. Therefore, bounds on the motor current to prevent overheating translate into bounds on the motor torques

$$\tau_i^{\min} \le \tau_i \le \tau_i^{\max}, \quad i = 1, 2, \ldots, m, \tag{3}$$

where $\tau_i^{\min} < 0$, $\tau_i^{\max} > 0$ represent the lower and upper bounds, respectively. Given the model (1), the constraints (3) translate into the following constraints

$$\ddot{\Theta}_i^{\min} \le \ddot{\Theta}_i \le \ddot{\Theta}_i^{\max}, \ \ddot{\Theta}_i^{\max} = \frac{\tau_i^{\max}}{M_i}, \ \ddot{\Theta}_i^{\min} = \frac{\tau_i^{\min}}{M_i} \tag{4}$$

The following constraints relate either to the maximum voltage of the electrical motor supply or to mechanical constraints of the robot

$$\dot{\Theta}_i^{\min} \le \dot{\Theta}_i \le \dot{\Theta}_i^{\max} \tag{5}$$

Both of them limit the link velocity as described by Eq. (5). Usually, the upper and lower bounds in Eqs. (3) and (5) are equal but opposite

$$|\tau_i^{\min}| = \tau_i^{\max}, \quad i = 1, 2, \ldots, m, \tag{6}$$

$$|\dot{\Theta}_i^{\min}| = \dot{\Theta}_i^{\max} \tag{7}$$

According to (4), Eq. (6) translates into

$$|\ddot{\Theta}_i^{\min}| = \ddot{\Theta}_i^{\max} \tag{8}$$

The robot models (1)–(8) will be experimentally validated in Section 3 on the Eshed Robotec MK2 robot. In Section 2.2, we will not consider the constraints (5) and (7).

2.2. Minimum-time kinematic path planning without link velocity constraints

A kinematic path planner such as Ariadne's Clew Algorithm (Bessiere et al., 1995; Mazer et al., 1998), which we will apply because of its excellent computational efficiency, produces a set of n points in configuration space

$$\{\Theta^1, \Theta^2, \ldots, \Theta^n\}, \; \Theta^k \in R^{m\times 1}, \; k = 1, 2, \ldots, n, \tag{9}$$

where each vector $\Theta^k \in R^{m\times 1}$, $k = 1, 2, \ldots, n$ contains the associated link rotations Θ_i^k, $i = 1, 2, \ldots, m$, respectively. The kinematic planner checks whether collisions occur by linearly interpolating between these points. It checks whether the straight lines

$$\pi^k(\lambda) = \Theta^k + \lambda(\Theta^{k+1} - \Theta^k), \; 0 \le \lambda \le 1, \quad k = 1, 2, \ldots, n-1 \tag{10}$$

in configuration space are collision-free. To follow these straight lines, the robot has to stop at each point Θ^k, $k = 1, 2, \ldots, n$ because the direction of motion is discontinuous at these points, unless two consecutive straight lines point in the same direction. This is an important disadvantage of our method and it is the price we have to pay for using computationally very efficient kinematic planners with collision detection based on linear interpolation.

So, based on kinematic planning, we obtain a collision-free path consisting of the set of subsequent lines (10) in the configuration space. We will now show that given the dynamic models (1)–(3) and (6) of the robot, a motion along these straight lines is actually the fastest way to arrive from each point Θ^k to the next point Θ^{k+1}, $k = 1, 2, \ldots, n-1$ in configuration space, assuming that the robot stops at each point. So assuming the robot stops at each point, following the straight lines results in minimum-time transitions.

To demonstrate this, first consider the minimum time required by a single robot link i to move from one location Θ_i^k to the next Θ_i^{k+1}, $k = 0, 1, \ldots, n-1$ without loss of generality assume $\Theta_i^{k+1} - \Theta_i^k > 0$. Then due to the fact that the initial and final velocity have to be zero and due to the absence of friction the control and motion required for a transition in minimum time are given by

$$\begin{aligned} \tau_i(t) = \tau_i^{\max}, \; \dot{\Theta}_i(t) = \ddot{\Theta}_i^{\max} t, \quad 0 \le t \le t_{i,s}^k, \\ \tau_i(t) = -\tau_i^{\max}, \quad \dot{\Theta}_i(t) = \ddot{\Theta}_i^{\max}(t_{i,f}^k - t), \quad t_{i,s}^k \le t \le t_{i,f}^k, \end{aligned} \tag{11}$$

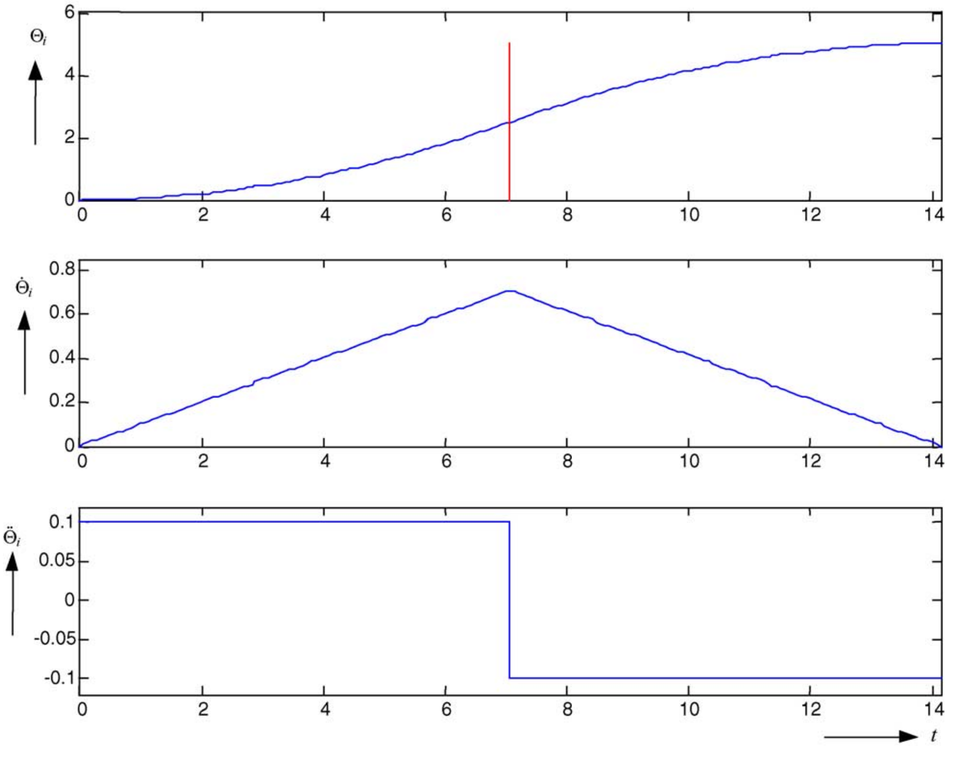

Fig. 1. Minimum-time motion profiles for $\Theta_i^{k+1} - \Theta_i^k = 5,\ \ddot{\Theta}_i^{\max} = 0.1$.

where 0, $t_{i,s}^k$, and $t_{i,f}^k$ denote, respectively, the initial time, the time at which the control variable switches from the maximum to the minimum value, and the final time of the minimum time motion of the i^{th} link from Θ_i^k to Θ_i^{k+1}. Fig. 1 represents this minimum-time motion. For obvious reasons, the control (11) is called a bang-bang control.

Given the symmetry of the bounds (8) for the bang-bang control and motion, one easily computes that

$$t_{i,s}^k = \sqrt{\frac{\Theta_i^{k+1} - \Theta_i^k}{\ddot{\Theta}_i^{\max}}}, \quad t_{i,f}^k = 2t_{i,s}^k \tag{12}$$

Because the motions of the links are independent the minimum time to move from Θ^k to Θ^{k+1} equals

$$t^k = \max_i (t_{i,f}^k) \tag{13}$$

The value i for which the maximum is attained in Eq. (13) is the index of the so-called ‘limiting link’ since this link requires the longest time to move from Θ_i^k to Θ_i^{k+1}. The control of this link is of the bang-bang type (11). Note that in exceptional cases, the maximum may be attained by more than one link. In that case, all these links are limiting. Also note that in the case of more than one limiting link, the control of each link is of the bang-bang type (11) and so the control of each limiting link switches at the same time. Due to this, if we consider only the motion of these limiting links in the configuration space, it would be a straight line. And this is exactly what we want since the collision avoidance is checked among the straight line in configuration space from Θ^k to Θ^{k+1}.

The remaining question is how the non-limiting links must be controlled. Because they are non-limiting their control can be selected in many different ways leading to different motions in the configuration space. From the discussion concerning the case of several limiting links it follows that (1) we want a motion in configuration space along a straight line and (2) this is automatically achieved if all the links would be limiting. The latter presents the answer of how to control the non-limiting links. We have to scale down their maximum motor torque to exactly the point where they become 'limiting'. At this point their control must be of the bang-bang type (11) again resulting in a motion in configuration space along a straight line. So the remaining question is how much the torques of the non-limiting links should be scaled down. Given the bang-bang control (11) the traveled distance $\Theta_i(t) - \Theta_i^k$ of each link at each time $0 \leq t \leq t_{i,f}^k$ is proportional to the scaled maximum torque. Therefore, if link j is a limiting link then the scaled down maximum torques $\tau'^{\max}_i$ of the other links should satisfy

$$\tau'^{\max}_i = \frac{\Theta_i^{k+1} - \Theta_i^k}{\Theta_j^{k+1} - \Theta_j^k} \tau_j^{\max} \leq \tau_i^{\max} \tag{14}$$

The inequality in Eq. (14) holds due to the fact that the jth link is limiting. This concludes the minimum-time path planning and associated control computation based on Ariadne's Clew Algorithm when no link velocity constraints are considered. In the next section these will be taken into account.

2.3. Minimum-time kinematic path planning including link velocity constraints

As in Section 2.2, we will start to consider the minimum-time control and motion of a single link given the presence of the additional constraints (5) and (7)

$$\begin{aligned}
&\tau_i(t) = \tau_i^{\max},\ \dot{\Theta}_i(t) = \ddot{\Theta}_i^{\max} t,\ 0 \leq t < t_{i,s}^k \\
&\tau_i(t) = 0,\ \dot{\Theta}_i(t) = \ddot{\Theta}_i^{\max} t_{i,s}^k,\ t_{i,s}^k \leq t \leq t_{i,f}^k - t_{i,s}^k \\
&\tau_i(t) = -\tau_i^{\max},\ \dot{\Theta}_i(t) = \ddot{\Theta}_i^{\max}(t_{i,f}^k - t),\ t_{i,f}^k - t_{i,s}^k < t \leq t_{i,f}^k
\end{aligned} \tag{15}$$

where $t_{i,s}^k$ represents the time instant where the control switches to zero. From Eqs. (8), (12) and (15), we obtain the following outcome of $t_{i,s}^k$ and $t_{i,f}^k$ for a minimum-time motion. If

$$\frac{\dot{\Theta}_i^{\max}}{\ddot{\Theta}_i^{\max}} \geq \sqrt{\frac{\Theta_i^{k+1} - \Theta_i^k}{\ddot{\Theta}_i^{\max}}} \tag{16}$$

then,

$$t_{i,s}^k = \sqrt{\frac{\Theta_i^{k+1} - \Theta_i^k}{\ddot{\Theta}_i^{\max}}}, \quad t_{i,f}^k = 2t_{i,s}^k \tag{17}$$

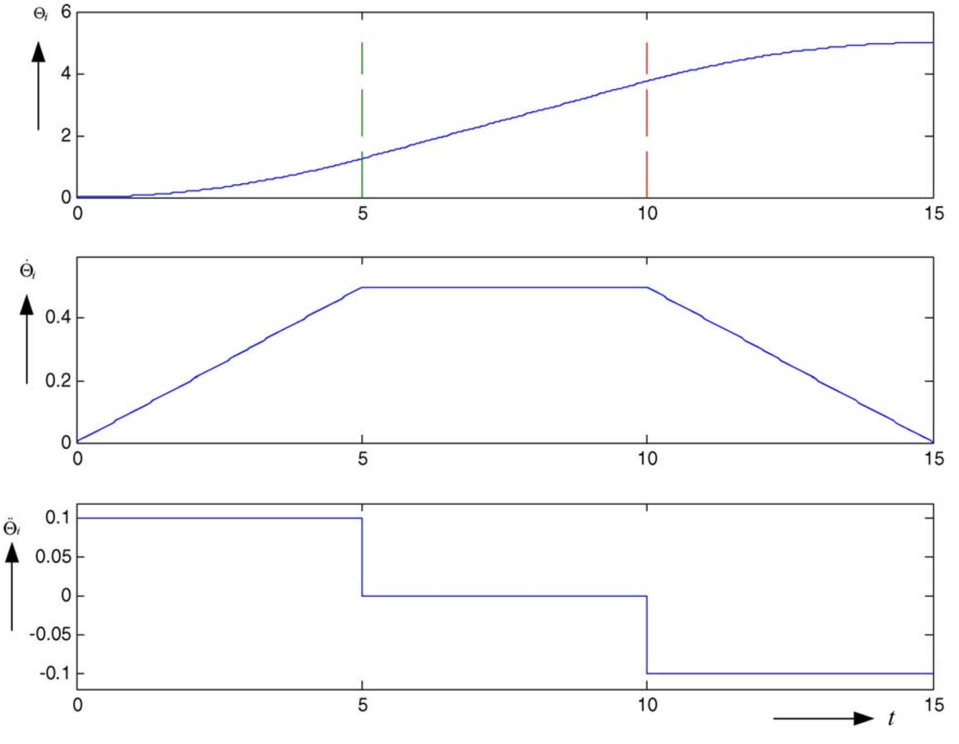

Fig. 2. Minimum-time motion profiles for $\Theta_i^{k+1} - \Theta_i^k = 5,\ \dot{\Theta}_i^{\max} = 0.5,\ \ddot{\Theta}_i^{\max} = 0.1$.

otherwise,

$$t_{i,s}^k = \frac{\dot{\Theta}_i^{\max}}{\ddot{\Theta}_i^{\max}}, \quad t_{i,f}^k = \frac{\Theta_i^{k+1} - \Theta_i^k}{\dot{\Theta}_i^{\max}} + t_{i,s}^k = \frac{\Theta_i^{k+1} - \Theta_i^k}{\dot{\Theta}_i^{\max}} + \frac{\dot{\Theta}_i^{\max}}{\ddot{\Theta}_i^{\max}} \tag{18}$$

If (16) applies, the maximum velocity is not reached and the control is bang-bang and equal to the one in the previous section. If (16) does not apply, the maximum velocity is reached and for obvious reasons the control (15) is called a bang-off-bang control. Observe that when in (16) the equality holds Eqs. (17) and (18) are identical and furthermore that (16) is equivalent with

$$\frac{\dot{\Theta}_i^{\max}}{\ddot{\Theta}_i^{\max}} \geq \frac{\Theta_i^{k+1} - \Theta_i^k}{\dot{\Theta}_i^{\max}} \tag{19}$$

Using Eq. (13) from Eqs. (16)–(18), the limiting link is easily computed. Fig. 2 represents a minimum-time motion which violates (19) or equivalently (16), implying that the maximum velocity is reached.

When (19) applies, the situation is identical to the one in the previous section. Therefore, we will now consider only the situation in which (19) does not apply, i.e. when (18) applies.

The maximum distance traveled by each link $i = 1,2, \ldots, m$ when the control (15) is applied equals

$$\Theta_i^{\max} = \ddot{\Theta}_i^{\max}(t_{i,s}^k)^2 + \dot{\Theta}_i^{\max}(t_{i,f}^k - 2t_{i,s}^k) \tag{20}$$

To achieve a motion along a straight line in configuration space, the time instants $t_{i,s}^k$ in Eq. (15) must be identical for each $i = 1,2, \ldots, m$ and also the time instants $t_{i,f}^k$. Observe from Eq. (20) that when the control (15) is applied, the traveled distance of each link $i = 1,2, \ldots, m$ is proportional to both $\dot{\Theta}_i^{\max}$ and $\ddot{\Theta}_i^{\max}$. Note from Eq. (4) that $\ddot{\Theta}_i^{\max}$ is proportional to $\tau_i^{\max}$. Therefore, if the jth link is limiting to move in minimum-time along a straight line in configuration space using the control (15), similar to Eq. (14), we have to scale $\tau_i^{\max}$, $\dot{\Theta}_i^{\max}$ into $\tau_i'^{\max}$, $\dot{\Theta}_i'^{\max}$

$$\tau_i'^{\max} = \frac{\Theta_i^{k+1} - \Theta_i^k}{\Theta_j^{k+1} - \Theta_j^k}\tau_j^{\max}, \quad \dot{\Theta}_i'^{\max} = \frac{\Theta_i^{k+1} - \Theta_i^k}{\Theta_j^{k+1} - \Theta_j^k}\dot{\Theta}_j^{\max} \tag{21}$$

Unfortunately, now we can no longer guarantee that the scaling (21) does not violate the constraints (3) and (5). The reason is that from (18) the limiting link is determined by a combination of the two bounds $\dot{\Theta}_i^{\max}$ and $\tau_i^{\max}$, which is proportional to $\ddot{\Theta}_i^{\max}$. Therefore, after scaling the individual bounds may be violated.

Suppose

$$\frac{\dot{\Theta}_i^{\max}}{\ddot{\Theta}_i^{\max}} = \frac{\dot{\Theta}_i^{\max} M_i}{\tau_i^{\max}} = c, \quad i = 1, 2, 3, \ldots, m \tag{22}$$

where c is a positive constant. Eq. (22) implies that all links reach their associated maximum velocities at the same time when they are maximally accelerated from zero. Observe that Eqs. (18), (21) and (22) imply

$$\frac{\tau_i'^{\max}}{\tau_i^{\max}} = \frac{\ddot{\Theta}_i'^{\max}}{\ddot{\Theta}_i'^{\max}} = \frac{\dot{\Theta}_i'^{\max}}{\dot{\Theta}_i^{\max}} \le 1, \quad i = 1, 2, \ldots, m \tag{23}$$

Eq. (23) implies that Eq. (22) is a sufficient condition to guarantee that the scaling (21) does not violate the constraints (3) and (5). In practice, robots are usually designed and build such that Eq. (22) is approximately satisfied. Therefore the violation of the bounds by the scaling (21) is unlikely. This is confirmed in the next section if we disregard the motion of the last joint which represents the gripper rotation. If the violation of the bounds does occur a not necessarily optimal solution would be to further scale down all the maximum velocities and accelerations.

3. Implementation on a five degrees-of-freedom robot

The five degrees of freedom Eshed Robotec MK2 robot is used to implement and experimentally verify our near minimum-time control approach. This robot is intended for the picking of tomatoes. The experiment described in this section relates to this task. The relevant technical information concerning this robot can be found in Tables 1 and 2, where

Table 1
The Denavit Hartenberg parameters, as defined in Craig (1986), of the MK2 robot

i	α_i (degrees)	a_{i-1} (m)	d_i (m)	Θ_i (degrees)
1	0	0	0	−152.5 152.5
2	−90	0.2	0	−125 30
3	0	0.27	0	−117.5 117.5
4	0	0.23	0	−22.5 202.5
5	90	0.15	0	−180 180

Table 2
Gear ratios, maximum accelerations, and velocities of the MK2 robot links

i	Gear ratio	$\ddot{\Theta}_i^{max}$ (rad/s^2)	$\dot{\Theta}_i^{max}$ (rad/s)	$\dot{\Theta}_i^{max}/\ddot{\Theta}_i^{max}$ (s)
1	1:120	290	150	1.93
2	1:160	285	90	3.16
3	1:144	304	140	2.17
4	1:120	476	180	2.64
5	1:88	100	476	0.2

the symbols and description comply with those in Craig (1986). The gear ratio given in Table 2 is the ratio between the angular velocity of the link and the angular velocity of the motor driving the link, as a consequence of the gear system in each link of the robot. The control was performed by individual PID controllers for each link feeding back the angular

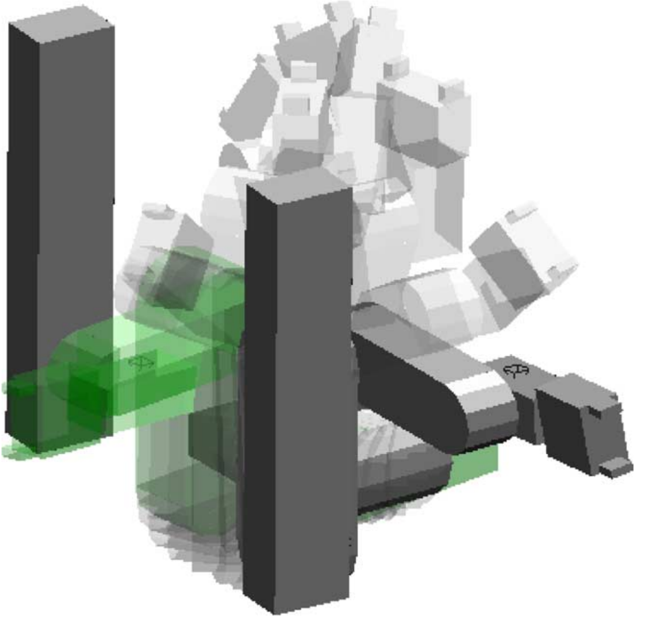

Fig. 3. Stroboscopic view of the path planned by Ariadne's Clew Algorithm. The two vertical blocks represent plant obstacles which are avoided.

positions of each link measured through optical encoders. The maximum velocities were specified by the manufacturer while we performed single link experiments to determine their maximum acceleration.

The MK2 robot controller can be programmed by specifying a sequence of points in the configuration space and by specifying how long the motion along these points should take. The robot controller interpolates between these points in configuration space and furthermore assumes that the velocity at the initial and final point should be zero. Furthermore, the robot controller is able to execute several of these motions, one after the other. This enables implementation of our control approach without any modifications on the robot controller. Several points along the straight lines in configuration space, computed in Sections 2.2 and

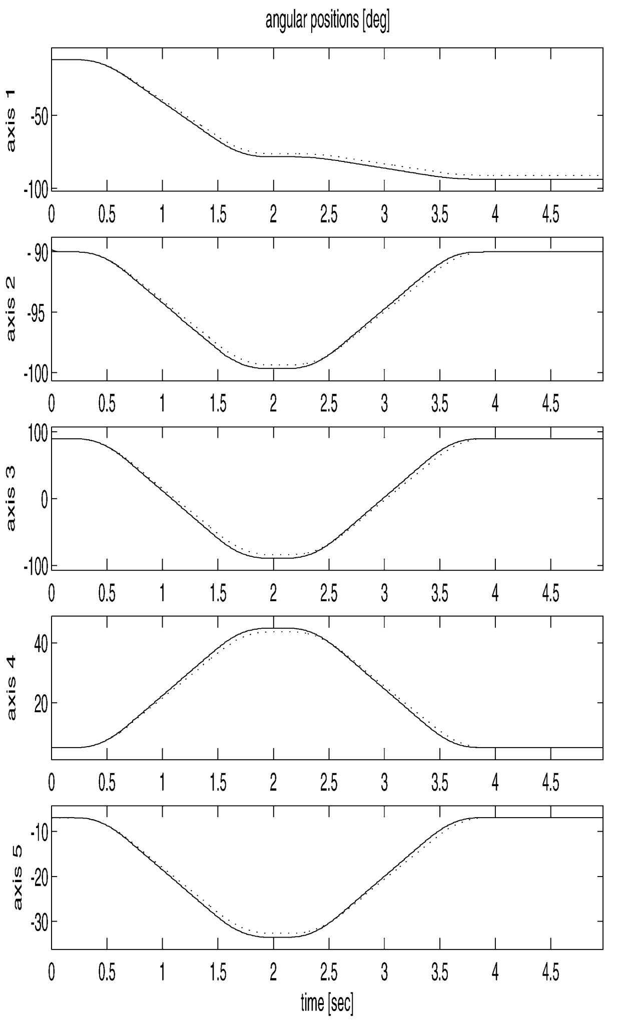

Fig. 4. Measured (−) and computed (:) joint angles.

2.3, are supplied to the robot controller together with the associated motion time which was also computed in Sections 2.2 and 2.3. In case of a sequence of straight lines, the robot then stops at the intermediate points connecting one line to the next. Experiments revealed that in addition to the specified motion time, the robot takes about 0.2 s to exactly reach the final position and the zero velocity. So, each stop at an intermediate point, connecting one line to the next, adds another 0.2 s to the computed motion time in Sections 2.2 and 2.3.

The following single experiment was performed. Ariadne's Clew Algorithm was used to solve a path planning problem involving two obstacles representing two tomato plants standing next to each other, as is common practice in greenhouses (see Fig. 3). The representation of the robot and obstacles in 3D, demanded by Ariadne's Clew Algorithm, is

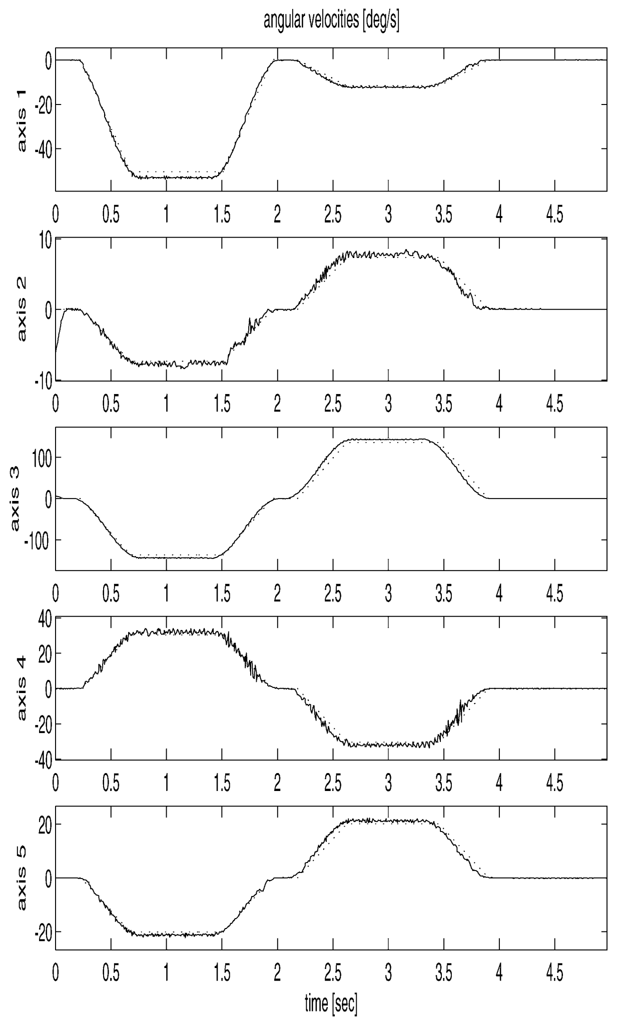

Fig. 5. Measured (−) and computed (:) joint velocities.

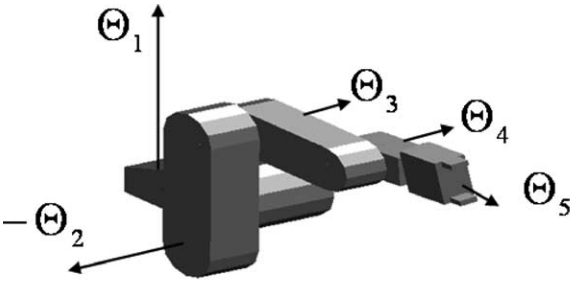

Fig. 6. Definitions of the joint angles.

through cylinders, the top and bottom of which are polygons (Bessiere et al., 1995; Mazer et al., 1998). The approximation of the robot and the obstacles by these cylinders is seen in Fig. 3. Given the approximation of the robot and obstacles by these cylinders, Ariadne's Clew Algorithm computes truly collision-free paths. For our example, on an ordinary PC, the algorithm computed a collision-free path within 15 s. This is orders of magnitude faster than the results reported by Van Henten et al. (2003b) obtained for a similar example using the A*-search algorithm. The path is represented by an intermediate point between the start and endpoint, such that the two associated straight lines in configuration space, connecting these points, are collision-free. The associated minimum-time position and velocity profiles of each link, computed by our algorithm, are displayed in Figs. 4 and 5 together with the measured profiles during the control experiment. For clarity the definitions of the angles of the robot links are shown in Fig. 6. From Fig. 5 it is obvious that velocity bounds are present and active. Therefore the computations are those described in Section 2.3 which took only 0.03 seconds on the same PC. Observe from Figs. 4 and 5 that the robot stops after approximately 2 seconds at the intermediate point connecting the two straight lines in configuration space, which avoid the two obstacles. The presence of the two obstacles is also apparent from the joint angle profiles in Fig. 4. Note that the initial and final angle of each joint are equal except for joint 1. Despite this all joints rotate significantly to avoid the two obstacles. Taking into account the additional 0.2 s taken by the robot controller to achieve the intermediate stop, the robot motion was computed to take 3.68 s. The actual motion took 3.62 s. This result illustrates the feasibility and accuracy of our approach in actual practice, which does not require any modification of the robot controller.

4. Conclusions

The method presented has been shown to compute effectively near minimum-time robot motions amongst obstacles. The efficiency of our computational method makes it a feasible for on-line path planning of fruit-picking robots. Note from the experiment in Section 3 that using our PC, and the kinematic planner, the computation time of this planner still exceeds the actual motion time required by the robot by about a factor 4. Clearly a computer that would be an order of magnitude faster would bring the computation time within the actual motion time needed by the robot.

In addition to the use of kinematic path planners, the contribution of this paper is to compute how, and under which simplifying conditions relating to the robot dynamics and constraints, the paths generated by kinematic planners are turned into near minimum-time motions. The simplifying assumptions are that the motions of the robot links are decoupled, that the bounds on the motor torques and velocities are symmetric and that the dynamics of each link can be described by essentially a double integrator. Abandoning one of these assumptions causes the paths generated by kinematic planners, being straight lines in configuration space, not to be time-optimal any longer.

The highest price we have to pay for being able to use the paths generated by kinematic planners is the need to stop at the intermediate points generated by these planners. But in the case of on-line path planning, given the current speed of computers, the gain in computation time easily outweighs the loss in motion time. Nevertheless a way to relax the problem might be to enlarge the objects representing the obstacles, which allows us afterwards to smooth the computed path, without introducing collisions. The smoothed path will no longer require the robot to stop. In future research we will apply this approach to our fruit-picking robot.

In the case of on-line path planning, as long as the computation time exceeds or equals the actual motion time of the robot, simple near time-optimal solutions, like the one in this paper, should be preferred over more general and advanced approaches being approximately or truly time-optimal.

References

Arima, S., Kondo, N., 1999. Cucumber harvesting robot and plant training system. Journal of Robotics and Mechatronics 11 (3), 208–212.

Bessiere, P., Ahuactzin, J.M., Talbi, E.G., Mazer, E., 1995. The Ariadne's Clew Algorithm: Global Planning With Local Methods. Algorithmic Foundations of Robotics, pp. 39–47.

Bobrow, J.E., Dubowsky, S., Gibson, J.S., 1985. Time-optimal control of robotic manipulators along specified paths. International Journal of Robotics Research 4 (3), 3–17.

Craig, J.J., 1986. Introduction to Robotics Mechanics and Control. Addison-Wesley.

Geering, H.P., Guzella, L., Hepner, S.A.R., Onder, C.H., 1986. Time-optimal motions of robots in assembly tasks. IEEE Trans. Aut. Cont. 31 (6), 512–518.

Hayashi, S., Sakaue, O., 1996. Tomato harvesting by robotic system. ASAE Annual International Meeting. Phoenix, Arizona, USA, ASAE Paper no. 96–3067.

Hol, C.W.J., van Willigenburg, L.G, van Henten, E.J., van Straten, G., 2001. A new optimization algorithm for singular and non-singular digital time-optimal control of robots, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol. 2, May 21–26. Seoul, Korea, pp. 1136–1141.

Kondo, N., Monta, M., Fujiura, T., 1996. Fruit harvesting robots in Japan. Advances in Space Research 18 (1/2), 181–184.

Mazer, E., Ahuactzin, J.M., Bessiere, P., 1998. The Ariadne's Clew Algorithm. Journal of Artificial Intelligence 9, 295–316.

Meier, E.B., Bryson, A.E., 1987. An efficient algorithm for time-optimal control of a two link manipulator, Proceedings of the AIAA Conference on Guidance Control (IEE). Montery, CA, pp. 204–212.

Ryuh, Y.S., Ryu, K.H., 1995. Obstacle-avoidance algorithm for fruit harvesting robots, Proceedings of the International Symposium on Automation and Robotics in Bioproduction and Processing, vol. 2, 3–6 November 1995. Kobe, Japan, pp. 1–8.

Shin, K.G., McKay, N.D., 1985. Minimum time control of robotic manipulators with geometric path constraints. IEEE Transactions on Automatic Control 30 (6), 531–541.

Shin, K.G., McKay, N.D., 1986. Selection of near minimum-time geometric paths for robot manipulators. IEEE Transactions on Automatic Control 31 (6), 501–511.

Van Henten, E.J., Hemming, J., Van Tuyl, B.A.J., Kornet, J.G., Meuleman, J., Bontsema, J., van Os, E.A., 2002. An autonomous robot for harvesting cucumbers in greenhouses. Autonomous Robots 13, 241–258.

Van Henten, E.J., Van Tuijl, B.A.J., Hemming, J., Kornet, J.G., Bontsema, J., Van Os, E.A., 2003a. Field test of an autonomous cucumber picking robot. Biosystems Engineering, in press, doi:10.1016/j.biosystemseng.2003a.08.002.

Van Henten, E.J., Hemming, J., Van Tuijl, B.A.J., Kornet, J.G., Bontsema, J., 2003b. Collision-free motion planning for a cucumber picking robot. Biosystems Engineering 86 (2), 135–144.

Van Willigenburg, L.G., 1991. Computation of time-optimal controls applied to rigid manipulators with friction. International Journal of Control 54 (5), 1097–1117.

Van Willigenburg, L.G., 1993. Computation and implementation of digital time-optimal feedback controllers for an industrial X-Y robot subjected to path, torque and velocity constraints. International Journal of Robotics Research 12 (5), 420–433.

Zlajpah, L., 1996. On time-optimal path control of manipulators with bounded joint velocities and torques, Proceedings of the 1996 IEEE International Conference on Robotics Automation, vol. 7. pp. 785–797.