

Available at www.sciencedirect.comjournal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot

E.J. Van Henten^{a,b,*}, E.J. Schenk^c, L.G. van Willigenburg^d, J. Meuleman^e, P. Barreiro^f

^a Wageningen UR Greenhouse Horticulture, P.O. Box 644, NL-6700 AA Wageningen, The Netherlands

^b Farm Technology Group, Wageningen University, P.O. Box 17, NL-6700 AA Wageningen, The Netherlands

^c SBG Innovatie, Hoornseweg 22, NL-1775 RB Middenmeer, The Netherlands

^d Systems and Control Group, Wageningen University, Bornsesteeg 59, NL-6708 PD Wageningen, The Netherlands

^e Plant Research International, Droevendaalsesteeg 1, 6708 PB Wageningen, The Netherlands

^f Laboratorio de Propiedades Físicas y Tecnologías Avanzadas en Agroalimentación LPF-TAG, Departamento de Ingeniería Rural ETSIA, Madrid, Spain

ARTICLE INFO

Article history:

Received 28 May 2008

Received in revised form

12 January 2010

Accepted 27 January 2010

Published online 14 April 2010

The paper presents results of research on an inverse kinematics algorithm that has been used in a functional model of a cucumber-harvesting robot consisting of a redundant P6R manipulator. Within a first generic approach, the inverse kinematics problem was reformulated as a non-linear programming problem and solved with a Genetic Algorithm (GA). Although solutions were easily obtained, the considerable calculation time needed to solve the problem prevented on-line implementation. To circumvent this problem, a second, less generic, approach was developed which consisted of a mixed numerical-analytic solution of the inverse kinematics problem exploiting the particular structure of the P6R manipulator. Using the latter approach, calculation time was considerably reduced. During the early stages of the cucumber-harvesting project, this inverse kinematics algorithm was used off-line to evaluate the ability of the robot to harvest cucumbers using 3D-information obtained from a cucumber crop in a real greenhouse. Thereafter, the algorithm was employed successfully in a functional model of the cucumber harvester to determine if cucumbers were hanging within the reachable workspace of the robot and to determine a collision-free harvest posture to be used for motion control of the manipulator during harvesting. The inverse kinematics algorithm is presented and demonstrated with some illustrative examples of cucumber harvesting, both off-line during the design phase as well as on-line during a field test.

© 2010 IAGrE. Published by Elsevier Ltd. All rights reserved.

1. Introduction

In 1996, research was initiated at the former Institute of Agricultural and Environmental Engineering (IMAG) to develop an autonomous cucumber-harvesting robot with financial support from the Dutch Ministry of Agriculture,

Nature and Food Quality. The reduction of labour costs, problems with the availability of skilled labour and the need to improve the production process in a quantitative as well as a qualitative sense, were the main driving forces for automation. Various aspects of the development of this agro-robotic system such as the adoption of a new cultivation

* Corresponding author.

E-mail address: eldert.vanhenten@wur.nl (E.J. Van Henten).

1537-5110/\$ – see front matter © 2010 IAGrE. Published by Elsevier Ltd. All rights reserved.

doi:10.1016/j.biosystemseng.2010.01.007

Notation	
α	Link twist
θ	Joint angle
ρ	Polar coordinate radius
ϕ	Polar coordinate angle
a	Link length
d	Link offset
D	Shortest distance between manipulator and obstacle
J	Performance criterion
J_{pos}	Penalty on position deviations of TCP from target
J_{ori}	Penalty on orientation deviations of TCP from target
J_{dist}	Penalty accounting for distance between manipulator and obstacles
K	Intermediate variable
p	Position coordinate
R	Rotation matrix
r_{ij}	Element (i,j) of rotation matrix
T	Transformation matrix
<i>Superscript</i>	
*	Optimal value
<i>Subscripts</i>	
min	Lower bound
max	Upper bound
target	Object to be harvested
TCP	Tool centre point
x,y,z	x, y, z axes

system, economics, logistics and, last but not least, the robot technology have been reported already (Van Henten, Hemming, Van Tuijl, Kornet, & Bontsema, 2003; Van Henten et al., 2002; Van Willigenburg, Hol, & Van Henten, 2004). A field test of the robot was reported by Van Henten, Van Tuijl, et al. (2003). Optimal redesign of the manipulator of the harvester was addressed by Van Henten, Van Willigenburg, and Van't Slot (2009).

This paper focuses on one particular issue of the cucumber-harvesting robot that has not been reported so far, namely the calculation of the collision-free inverse kinematics of the manipulator. In the harvesting robot, inverse kinematics calculations are used for two purposes. Given the position of a harvestable cucumber, inverse kinematics calculations are, first of all, used to determine whether or not the cucumber lies within the reachable workspace of the manipulator. Skipping fruit lying outside the workspace saves costly operating time. Secondly, these calculations are used to derive a collision-free harvest configuration of the manipulator. Using information from the vision system about the position and orientation of the cucumber fruit as well as information about objects in the workspace, the end-effector should be positioned in such a way that neither the end-effector nor the manipulator collides with other objects in the workspace of the robot such as leaves, stems, other fruit, greenhouse construction components or the robot itself. In Dutch horticultural practise, robots have to operate in a tight working environment and collisions with the canopy and the greenhouse construction may have serious economic consequences and thus should be prevented. The calculated harvest configuration is then used as input for the derivation of a collision-free motion to steer the manipulator during the harvest process (see e.g. Van Henten, Hemming, et al., 2003; Van Willigenburg et al., 2004). As far as the authors are aware, the derivation of collision-free inverse kinematics of complex and redundant kinematic chains has received no explicit attention in the agricultural engineering world, though they play a crucial role during the design and operation of complex manipulator structures in agriculture.

The manipulator of the harvest machine involved in this study has seven links and consists of one prismatic joint and six rotational joints (P6R), has six degrees of freedom

(DOF) and thus is redundant. In practise, redundancy means that a multitude of manipulator postures yields the same position and orientation of the end-effector. The advantage of this redundant manipulator is that it is able to avoid collisions in a difficult and cluttered task environment. A distinct disadvantage of such a manipulator is the non-existence of an analytic solution of the inverse kinematics problem. Only for manipulators having a very particular geometric structure, satisfying the Pieper criterion, will the inverse kinematics problem have a closed form analytic solution (Chapelle & Bidaud, 2004; Craig, 1989). This is not the case for the manipulator of the cucumber harvester. Alternatively, solutions for the inverse kinematics problem can be generated using numerical methods relying on iterative procedures (Chapelle & Bidaud, 2004). Solution procedures based on non-linear programming such as those proposed by, for instance, Nearchou (1998), Parker, Khoogar, and Goldberg (1989) and Zhao and Badler (1994) are very attractive due to their generic nature. They can easily be adapted to different manipulator structures. In this research two different approaches have been used to solve the inverse kinematics problem. In a first generic approach, the inverse kinematics problem was transformed into a non-linear programming problem that was solved using a Genetic Algorithm (GA). The results were promising, though calculation time was found to be prohibitive for on-line application. Therefore a second, less generic, approach was developed. This approach exploits the particular structure of the manipulator leading in practise to a mixed numerical-analytic solution for the inverse kinematics within very limited computation time.

This paper has the following structure. In Section 2 the kinematic model of the harvesting robot is presented. In Section 3 a non-linear programming approach to solving the inverse kinematics problem for the P6R manipulator is described and results are presented and discussed. Section 4 describes the mixed numerical-analytic inverse kinematics algorithm. Section 5 describes three ways in which the mixed numerical-analytic inverse kinematics algorithm was used both during the development stage and during the actual operation of the cucumber harvester in a greenhouse. The paper ends with concluding remarks.

2. A kinematic model of the harvest robot

Fig. 1 shows a functional model of the cucumber-harvesting robot, an autonomous vehicle used for coarse positioning of the harvesting machine in the aisles of the greenhouse. This vehicle uses the heating pipes as a rail for guidance and support. It serves as a mobile platform for carrying power supplies, a pneumatic pump, various electronic hardware for data-acquisition and control, a camera vision system for detection and localisation of cucumbers and a seven-link manipulator for positioning of the end-effector. The manipulator consists of a linear slide on top of which a Mitsubishi RV-E2 manipulator with an anthropomorphic arm and a spherical wrist is mounted. The manipulator carries an end-effector. This end-effector contains two parts: a gripper to grasp the fruit and a cutting device to detach the fruit from the plant. For more constructional details of the harvest robot refer to Van Henten et al. (2002).

Fig. 2 shows the kinematic structure of the seven-link manipulator. This kinematic model was derived using the Denavit–Hartenberg convention described by Craig (1989). The Denavit–Hartenberg parameters of this manipulator are listed in Table 1. Using this kinematic model, a 3D-model of the harvest robot was constructed as shown in various figures throughout the paper. This model was used for visualisation purposes and served as a basis for a collision detection algorithm used during the calculation of the inverse kinematics.

3. A non-linear programming approach to solving the inverse kinematics problem

In this project a non-linear programming approach to solving the inverse kinematics problem was implemented for its generic nature so that it could easily be applied to manipulators with a different number of links and different configuration.

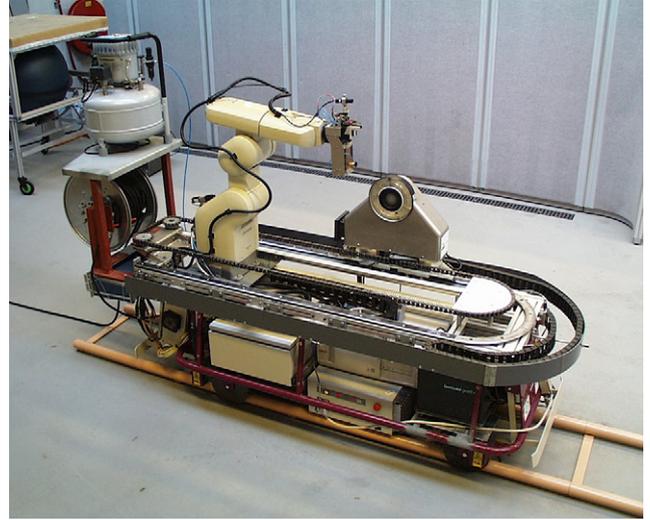


Fig. 1 – The cucumber-harvesting robot.

deviations of the tool centre point (TCP) from the desired harvesting position, a second term, J_{ori} , penalising deviations of the TCP orientation from the desired harvest orientation and a final term, J_{dist} , penalising collisions and close encounters between the manipulator and objects in the working environment:

$$J = J_{pos} + J_{ori} + J_{dist}. \quad (3)$$

Since most cucumbers have a more or less vertical orientation, the gripper used should have a vertical posture during harvesting as illustrated in Fig. 3. Therefore, it was assumed that four DOF, consisting of three translations and one rotation, would suffice to harvest a cucumber. Deviations of the TCP from such a harvest position were penalised with:

$$J_{pos}(d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = \sqrt{(p_{x,target} - p_{x,TCP})^2 + (p_{y,target} - p_{y,TCP})^2 + (p_{z,target} - p_{z,TCP})^2} \quad (4)$$

3.1. Methodology

For the P6R manipulator of the cucumber harvester, the non-linear programming approach boiled down to finding optimal values of the joint translation d_1 and six joint rotations $\theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ and θ_7 , satisfying

$$d_1^*, \theta_2^*, \theta_3^*, \theta_4^*, \theta_5^*, \theta_6^*, \theta_7^* = \inf_{d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7} J(d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \quad (1)$$

subject to constraints of the form

$$d_{1,min} < d_1 < d_{1,max}, \theta_{i,min} < \theta_i < \theta_{i,max}, \quad i = 2, \dots, 7 \quad (2)$$

in which $d_{1,min}$, $\theta_{i,min}$ and $d_{1,max}$, $\theta_{i,max}$ are lower and upper bounds on the joint parameters, respectively. The performance criterion J consisted of three terms, one, J_{pos} , penalising

in which $p_{x,target}$, $p_{y,target}$ and $p_{z,target}$ are the x, y and z coordinates of the cucumber and $p_{x,TCP}$, $p_{y,TCP}$ and $p_{z,TCP}$ represent the x, y and z co-ordinates of the TCP.

For the harvest posture shown in Fig. 3, the elements r_{11} , r_{21} , r_{31} , r_{32} and r_{33} of the rotation matrix $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ were required to take values $r_{11} = 0$, $r_{21} = 0$, $r_{31} = -1$, $r_{32} = 0$ and $r_{33} = 0$. Consequently, deviations of the TCP from this harvest orientation were penalised by

$$J_{ori}(d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = \sqrt{r_{11}^2 + r_{21}^2 + (1 + r_{31})^2 + r_{32}^2 + r_{33}^2}. \quad (5)$$

For collision prevention, the shortest distance D between the manipulator and the obstacles in the working environment was calculated using a bounding sphere algorithm

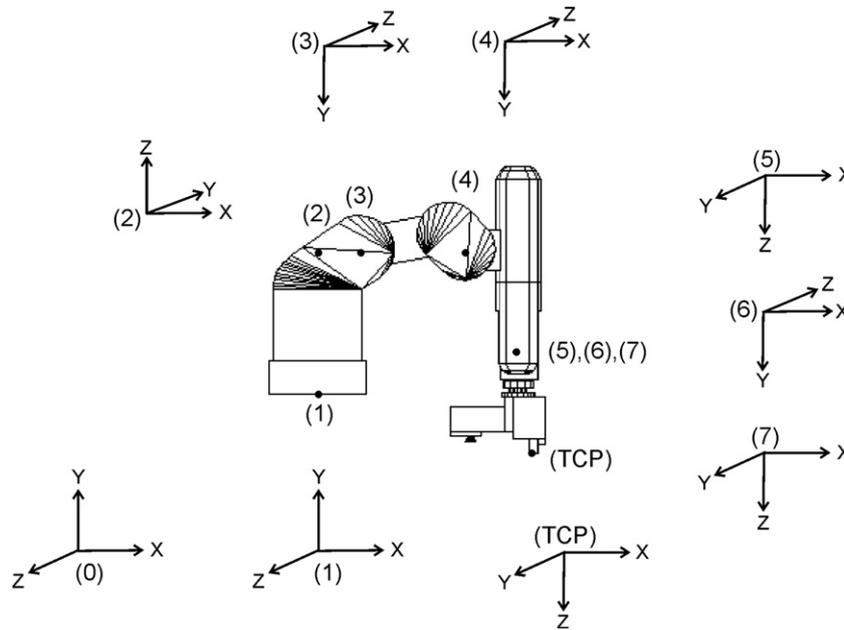


Fig. 2 – The kinematic structure of the seven-link manipulator of the cucumber-harvesting robot.

described in Appendix A. Using the aforementioned distance D , the following penalty was derived

$$J_{\text{dist}}(d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = \begin{cases} 1 & \text{if } D = 0 \text{ mm,} \\ 1 - D/75 & \text{if } 0 < D < 75 \text{ mm,} \\ 0 & \text{if } D > 75 \text{ mm.} \end{cases} \quad (6)$$

The non-linear programming problem was solved using the Matlab® Genetic Algorithm Toolbox.

3.2. Results

Fig. 4 shows two views of an artificial greenhouse environment including the harvesting robot, a cucumber to be harvested and several obstacles such as leaves represented by stars. In this scenery, a collision-free harvest posture of the manipulator was computed using the performance criteria defined above. Fig. 5 shows the results for two runs where the algorithm produced two feasible but clearly different postures. Both postures are indeed collision-free, however the posture in Fig. 5a has a considerably larger free space for manipulator manoeuvres than the posture shown in Fig. 5b. In terms of the performance criterion J , the solution shown in

Fig. 5a is better, having the lowest J value (0.0001467) which was obtained after 170 generations and 750 s on a state-of-the-art Pentium PC in the year 2000. The solution shown in Fig. 5b has a poorer performance value (0.0099359) obtained after 1500 generations and 6500 s of processing time.

3.3. Discussion

The main advantage of the non-linear programming approach to solving the inverse kinematics problem is its generic nature and flexibility in dealing with different manipulator structures. Programming the example presented above was

Table 1 – Link parameters and joint limits of the seven-link manipulator of the cucumber harvest robot.						
i	α_{i-1}	a_{i-1}	d_i	θ_i	Joint limit, min	Joint limit, max
1	0	0	d_1	0	0 mm	900 mm
2	-90°	0	350	θ_2	-160°	160°
3	-90°	100	0	θ_3	-45°	135°
4	0°	250	0	θ_4	50°	170°
5	-90°	130	250	θ_5	-160°	160°
6	90°	0	0	θ_6	-120°	120°
7	-90°	0	0	θ_7	-200°	200°

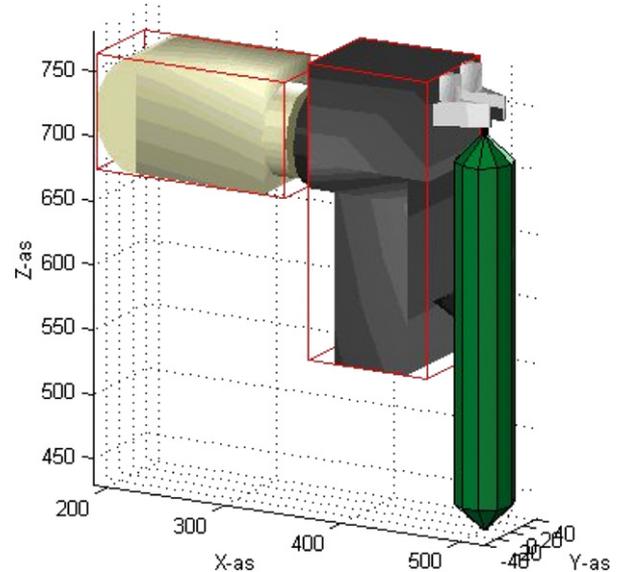


Fig. 3 – The vertical orientation of the cucumber reduces the required DOF from 6 (3 position and 3 orientation parameters) to 4 (3 position and 1 orientation parameters).

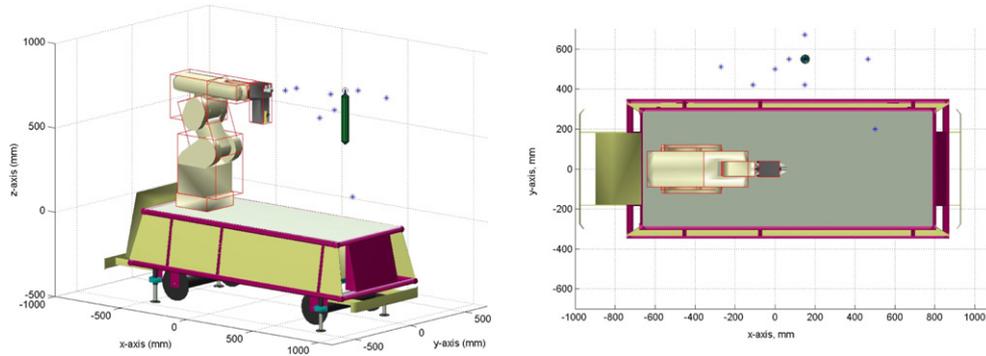


Fig. 4 – Two views of a virtual greenhouse workspace with a cucumber, miscellaneous objects like leaves denoted by blue stars, and the harvest robot. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

relatively easy, straightforward and fast. The research however, revealed two distinct disadvantages. First of all, the non-linear programming problem posed in this research appeared to be multi-modal. Despite the global search properties of the GA used, it got stuck in local minima during consecutive runs. This may have to do with the particular choice of the run-time parameters used in the GA and requires further investigation. A more important remark on this procedure is that it was found to be too slow for on-line implementation in the cucumber harvester. Solving the non-linear programming problem relies on a large number of time-consuming iterations. More computing power together with more efficient programming languages like C++ instead of Matlab® might improve the performance. In this project however, the computation time problem was circumvented by implementing a mixed numerical-analytic approach to solving the inverse kinematics problem for this particular manipulator as described in the next section.

4. A mixed numerical-analytic approach to solving the inverse kinematics problem

The second strategy to solving the inverse kinematics problem consisted of a mixed numerical-analytic approach

that exploited the particular kinematic structure of the manipulator used in the cucumber harvester in order to reduce the computation time.

4.1. Methodology

The manipulator of the harvesting robot consists of a Mitsubishi RV-E2 with six rotations mounted on top of a linear slide which introduces redundancy in the kinematic chain. The inverse kinematics of the isolated RV-E2 manipulator, however, can be solved analytically because the axes of the last three joints intersect. The addition of the linear slide introduces a twist, α_1 , and translation, d_1 , of the frame axes between the first and the second link. From the second link onwards, the kinematic chain is exactly the same as that of the RV-E2 manipulator. Therefore, following a similar procedure to that presented by Craig (1989) for a 5R PUMA manipulator, the inverse kinematics of the 7R manipulator were derived analytically from link 2 to link 7. This analytic solution is presented in Appendix B.

The addition of the linear slide introduces redundancy in the kinematic chain, and in most cases, an infinite number of values of d_1 within its physical bounds ($d_{1,min}, d_{1,max}$) allow the desired position and orientation of the TCP. To make the redundancy problem tractable, a practical approach to the solution of the inverse kinematics of the seven-link manipulator would be to

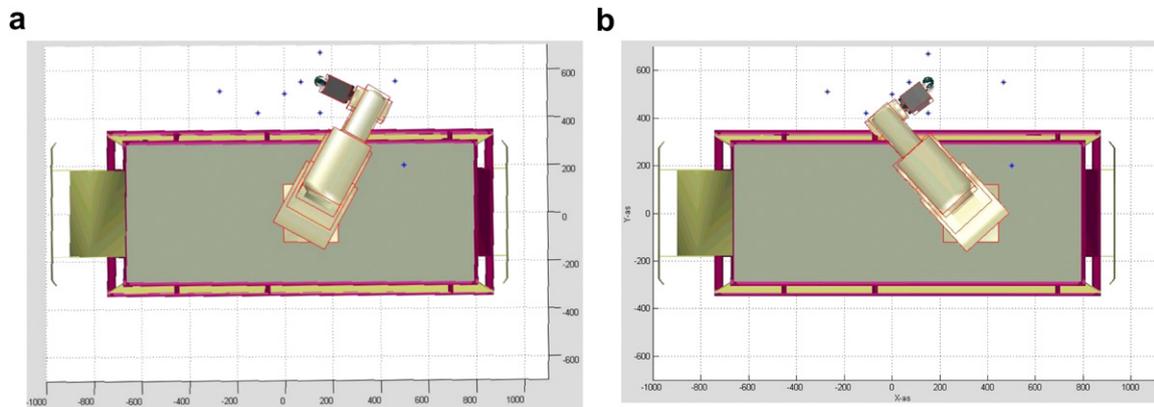


Fig. 5 – Two feasible solutions of the non-linear programming approach to solving the inverse kinematics problem; the blue stars indicate obstacles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

scan the translation d_1 in discrete steps between its physical bounds $d_{1,min}$ and $d_{1,max}$, to calculate the associated rotations θ_2 to θ_7 using the equations presented above and to choose one of the solutions. This choice might be based on the same criteria as mentioned in Section 3, including collision detection and distance between manipulator and objects in the workspace.

Based on this observation, a two-step approach was implemented. The first step consisted of establishing a proper angle for approaching the cucumber with the end-effector. This angle was found by scanning the workspace using discrete steps with the last three links of the robot, and selecting the best solution within the criterion of collision-free posture together with maximal distance between the three manipulator components and the surrounding obstacles. For the same artificial greenhouse environment shown in Fig. 4, this procedure is illustrated in Fig. 6. Collision detection and distance measurements were based on the bounding sphere approach as described in Appendix A. The best result is shown in Fig. 7. The second step consisted of deriving a configuration for the whole seven-link manipulator so as to achieve the position and orientation obtained during the first step. The final configuration was obtained by scanning the translation of the linear slide and by calculating the rotations of the 6 rotational joints using the equations presented above. Some intermediate results of this procedure are shown in Fig. 8. The choice of the best solution was based on a mixture of three criteria: collision-free posture, maximum distance between the manipulator and objects in the workspace, and, additionally, a maximum amount of freedom in the rotations of the 6 rotational joints. Results of the two-step discrete scanning process were tabulated, and a table search was used to find the solution best satisfying all criteria.

4.2. Results

The harvest posture calculated for the artificial greenhouse environment of Fig. 4, best satisfying all criteria, is shown in

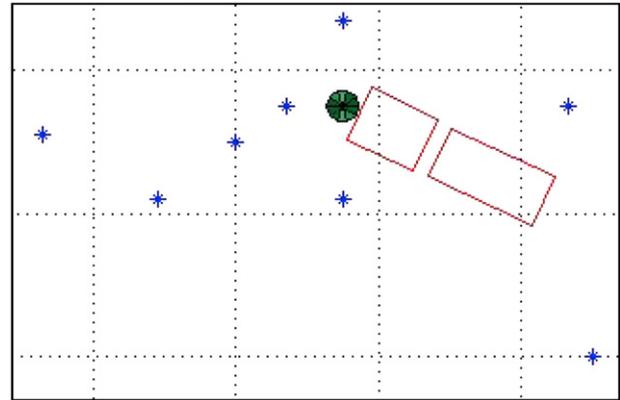


Fig. 7 – The angle of the last three links with the biggest distance between manipulator and objects in the workspace; the stars indicate obstacles.

Fig. 9. Observe that the solution very much resembles the solution obtained with the non-linear programming procedure shown in Fig. 5b, though a small difference between the solutions can be observed.

It is important to note that, in all cases considered with this algorithm, results were obtained within a second on a state-of-the-art PC in the year 2000.

4.3. Discussion

Without doubt the mixed numerical-analytic algorithm outperformed the non-linear programming algorithm in terms of calculation time. Calculation times were sufficiently small to allow for on-line implementation in the first functional model of the cucumber harvester (see Van Henten et al., 2002).

Clearly, this practical approach has some drawbacks as well. The approach taken is very specific for the type of manipulator used; it does not have a generic nature. It can be used for all P6R

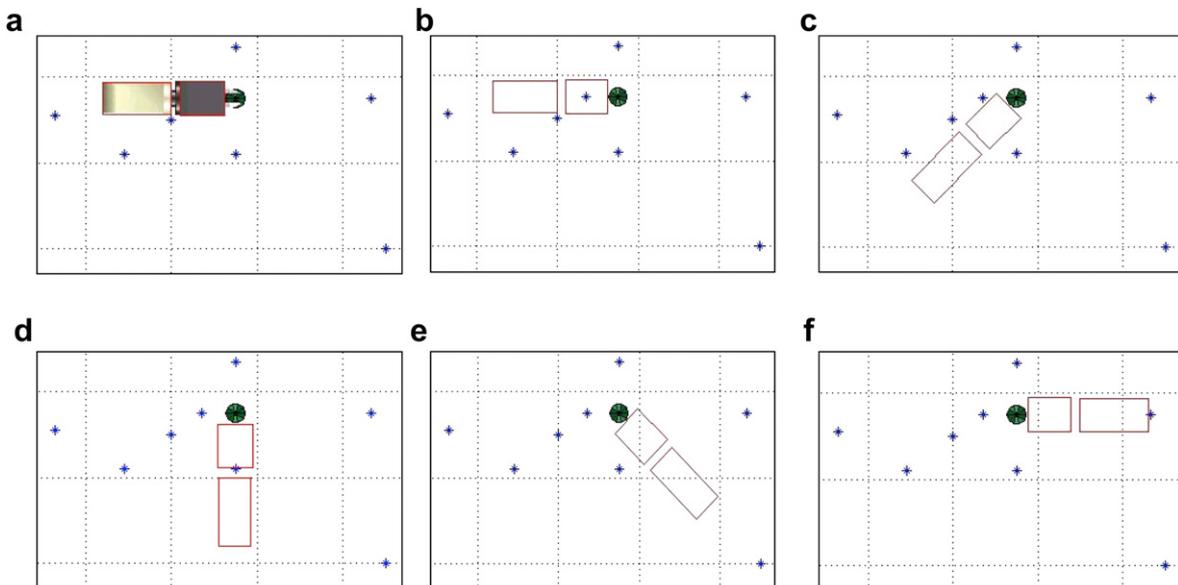


Fig. 6 – Scanning the angle of the last three manipulator links to obtain a harvest angle with the biggest distance between manipulator and objects in the workspace; the stars indicate obstacles.

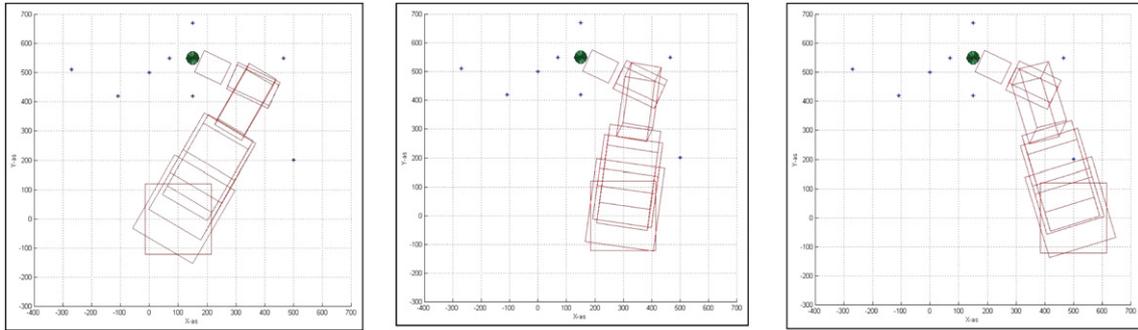


Fig. 8 – Scanning the linear slide in discrete steps to obtain the best manipulator posture achieving the predefined angle of the last three links, the largest distance between the manipulator and objects in the environment and most freedom of joint rotation; the stars indicate obstacles.

manipulators in which the 6R part satisfies the Pieper criterion, i.e. if the last three axes of the 6R manipulator intersect, an analytic solution of the inverse kinematics of the 6R manipulator can be obtained (Craig, 1989). Additionally, the algorithm does not yet include any recovery measures in case, after the discrete scanning steps, a feasible collision-free solution has not been obtained. In that case, the algorithm returns the answer that a solution cannot be found. But alternatively, the second best solution of the first step of the algorithm, with smaller distances to the obstacles might yield a feasible collision-free solution. These recovery measures as well as their importance under practical circumstances require further investigation.

5. Application of the inverse kinematics algorithm: some illustrative examples

The inverse kinematics algorithm was used in three applications. First of all, during the development phase of the robot, it was used to evaluate the ability of the robot to pick cucumbers under various realistic circumstances. Secondly, in the control

software of the functional model of the harvesting robot, the inverse kinematics algorithm was used to determine whether or not a cucumber was located within the (collision-free) workspace of the machine. Thirdly, if a cucumber was located within the workspace of the robot, the algorithm was used to calculate a collision-free harvest configuration which was used as the goal configuration for the robot motion planning algorithm.

5.1. Off-line evaluation of the robot during the design phase

During the design phase, the inverse kinematics algorithm was used for off-line evaluation of the ability of the robot to reach cucumbers in a real greenhouse environment. Some illustrative results will be presented here. Stereo images were taken using a standard Charged Coupled Device (CCD) camera in a greenhouse in which cucumbers were grown in a high-wire growing system (see Van Henten et al., 2002; Van Henten, Van Tuijl, et al., 2003). Using a manual feature-based matching technique, depth information was acquired from these two images. Not all pixel information was used.

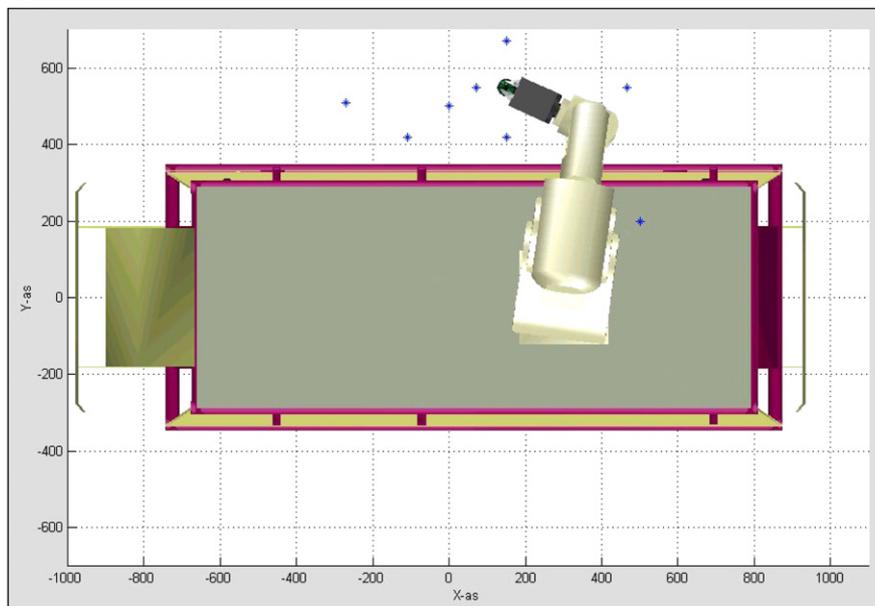


Fig. 9 – The mixed numerical-analytic solution of the inverse kinematics problem satisfying all criteria best.

Individual objects like leaves, stems and fruits were represented by an average of 15 points from which the distance to the camera information was calculated using a neural network. Using this information and the collision-free inverse kinematics procedure described above, harvest postures for the robot were calculated. Results are presented in Figs. 10 and 11. In both cases, harvesting the cucumbers was not trivial nor a guaranteed success. Leaves had to be removed to facilitate a collision-free harvesting posture. In the case of cucumber 1, indicated with C1, both a collision-free posture and a collision-free approach could be achieved after removing two leaves, indicated with an X, as shown in Fig. 10. In the case of cucumber 2, indicated with C2, harvesting was evidently much more difficult as shown in Fig. 11. The fruit was hidden in the crop between and behind leaves. To facilitate a collision-free harvesting posture a lot of obstacles had to be removed (indicated with X). It was concluded that a collision-free approach of this fruit could not be achieved with the current technology.

Based on a collection of these off-line simulation experiments it was concluded that, given the limited ability of current robot technology for automated harvesting, the

working environment had to be more structured and simplified by manually removing some of the leaves at the lower end of the plants. Fortunately, this already is a standard procedure with cucumbers grown in a high-wire cultivation system. Clearly, leaf removal is a time-consuming task. But in the mean time a first prototype leaf removal robot for cucumbers has been developed and tested as well (Van Henten *et al.*, 2006).

5.2. Field test of the cucumber-harvesting robot

The inverse kinematics algorithm was implemented in the research prototype of the cucumber harvester to determine whether or not a cucumber was located within the (collision-free) workspace of the machine. If a detected cucumber was not within the reachable workspace of the robot, the fruit would be skipped in the current detection and harvest cycle (see Van Henten, Van Tuijl, *et al.*, 2003 for more details on the operational procedure of the harvester). If a cucumber was located within the workspace of the robot, then secondly, the algorithm was used to calculate a collision-free harvest configuration which was used as the goal configuration for the

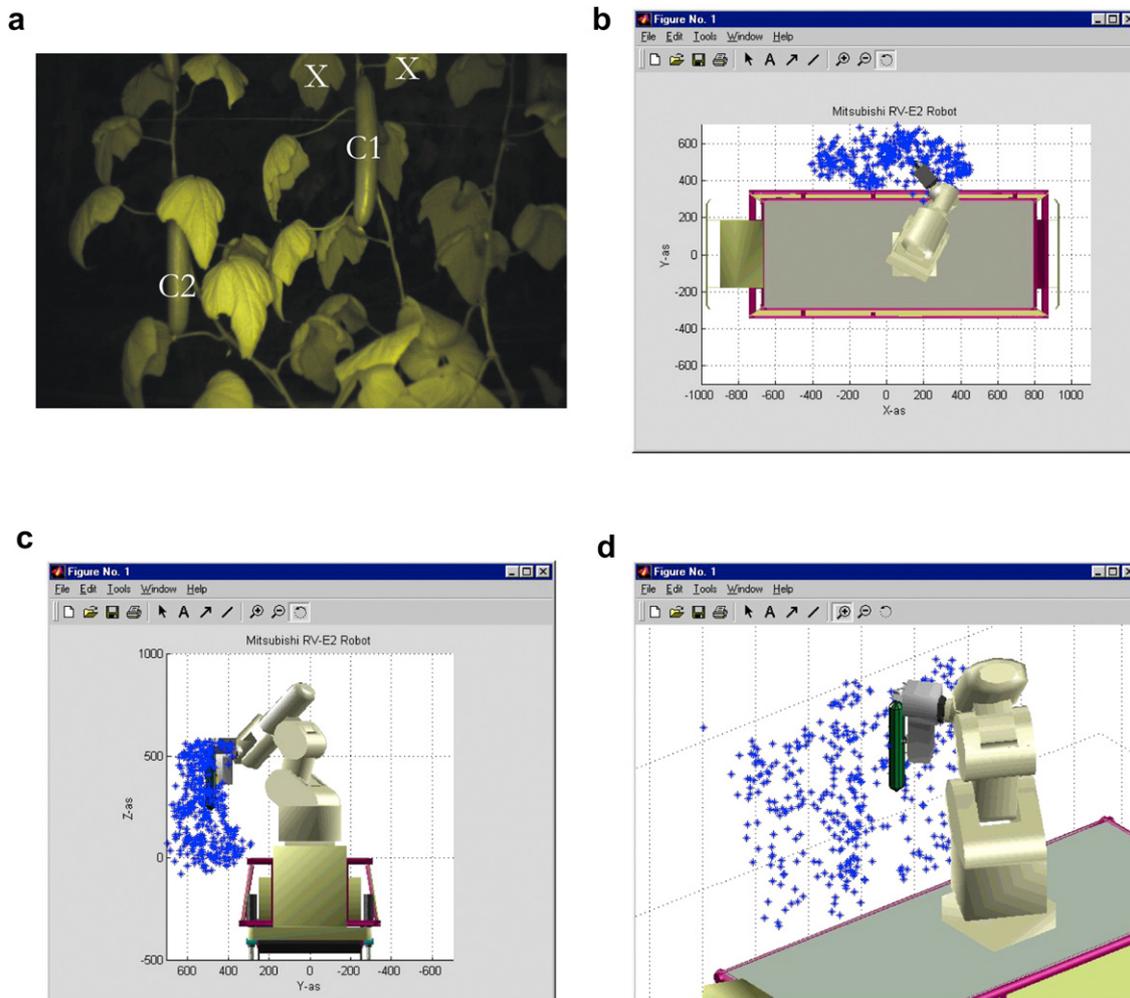


Fig. 10 – The original image (a) and three views (b), (c) and (d) of the manipulator posture to pick the cucumber indicated with C1; X indicate leaves that had to be removed to facilitate a collision-free posture, stars in (b)–(d) indicate obstacles.

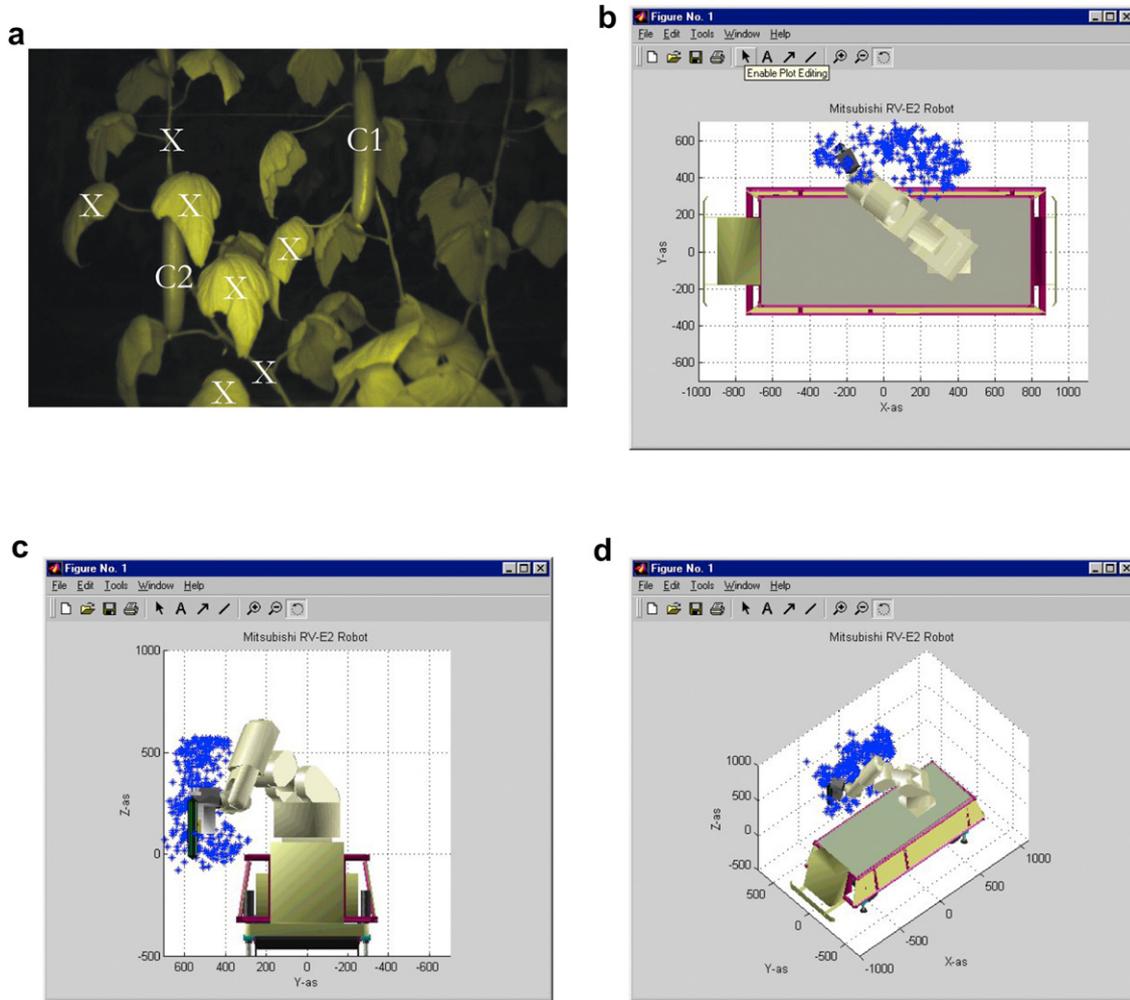


Fig. 11 – The original image (a) and three views (b), (c) and (d) of the manipulator posture to pick the cucumber indicated with C2; X indicate leaves that had to be removed to facilitate a collision-free posture, stars in (b)–(d) indicate obstacles.

collision-free manipulator motion planning algorithm as described by Van Henten, Hemming, *et al.* (2003). Detailed results of a field test of the cucumber-harvesting robot using this procedure are reported by Van Henten, Van Tuijl, *et al.* (2003). The success rate of the cucumber harvester was 74.4%. A single successful harvest cycle took 65.2 s per cucumber, but due to failed harvest attempts and the ability of the machine to perform at most three harvest attempts per cucumber, the success rate was improved and an average cycle time of 124 s per harvested cucumber was achieved. During the field test the performance of the inverse kinematics and motion planning software was evaluated in terms of time needed as well as failures. During a successful harvest attempt, calculation of the inverse kinematics and a collision-free motion as well as motion execution took approximately 12.5 s, *i.e.* 19% of the total execution time. The major portion of the 12.5 s was used for execution of the motion. The calculation of the inverse kinematics was not a limiting factor in this process. Of the failed harvest attempts, 4.9% could be attributed to motion planning and execution failures, including the harvest posture selection using the inverse kinematics algorithm. The majority of harvest failures however originated

from inaccurate positioning of the end-effector at the stalk of the fruit. These failures could be attributed mainly to errors in the detection and the accuracy of calculation of the 3D position of the cucumber. Combining the results of the inverse kinematics algorithm with the collision-free manipulator motion planning algorithm did not guarantee that collision-free paths could be calculated to feasible harvesting postures of the manipulator. Apparently, in this application, the relatively open structure of the crop allowed easy and collision-free access of the canopy and feasible harvesting postures could be reached with collision-free motions of the manipulator.

6. Final discussion, conclusions and directions for future research

In this paper a practical mixed numerical-analytic solution of the inverse kinematics problem for a redundant P6R manipulator was presented. Compared to a non-linear programming solution, this approach yielded feasible answers using very short computation times, which made

this approach practical for on-line implementation. However, the mixed numerical-analytic solution does not have the same generic nature of the non-linear programming approach and relies on the particular structure of the manipulator used. The current approach is feasible only for P6R manipulators of which the last six joints satisfy the Pieper criterion. The algorithm was successfully at two points in this robotics project. First of all, the algorithm was used during the early stages of robot development to evaluate the feasibility of picking cucumbers under real greenhouse environments. Based on these simulations and analyses, it was decided to modify the growing system to improve the feasibility of automated harvesting. Then, secondly, the algorithm was implemented in a functional model of the cucumber harvester and used to determine if the cucumbers were located within the workspace of the robot and if so, to produce a feasible harvest configuration to be used by the motion planning algorithm of the manipulator. During a field test experiment, overall motion planning and execution including the inverse kinematics calculations accounted for a minority of the failures and with a limited execution time, inverse kinematics calculations were not a bottle-neck in the harvest cycle.

Some issues require further research. Algorithms should be equipped with intelligent recovery procedures for those cases when the algorithm fails to generate a feasible answer, even though this answer may exist. Also, generation of collision-free harvest postures should be directly combined with the generation of collision-free manipulator motions to that posture, to assure that feasible postures can be reached with collision-free motions. In general this is not guaranteed and further research is needed to generate algorithms that are able to guarantee such results. Finally, a detailed error and sensitivity analysis might yield directions for further improvement of the algorithms described in this manuscript.

Acknowledgements

This work was financially supported by the Dutch Ministry of Agriculture, Nature and Food Quality. The authors gratefully acknowledge the constructive comments of two anonymous referees.

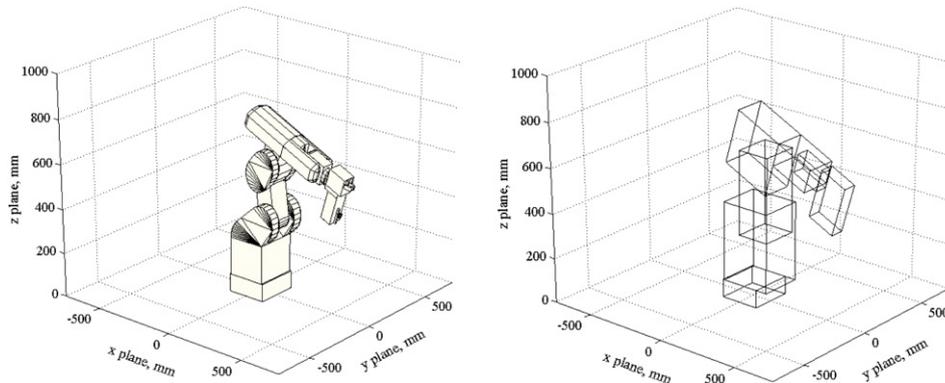


Fig. A.1 – Two 3D models of the Mitsubishi RV-E2 manipulator; (left) an accurate CAD model and (right) a less accurate model consisting of seven oriented bounding boxes.

Appendix A. Collision detection and distance estimation using bounding spheres

For collision detection and object to object distance calculation, a wide range of software packages is available (see Jiménez, Thomas, & Torras, 2001 for an overview). Still, accurate interference checking and distance calculation is a computationally intensive task. For practical applications, calculation time can be significantly reduced at the expense of a bit of accuracy. In this research, a fast but less accurate collision detection and distance calculation was based on a hierarchically-organised model representation of the robot using bounding spheres (see for instance Hubbard, 1996). The fundamental principle of collision detection using bounding spheres boils down to detecting whether or not the distance between the centres of two spheres is smaller than the sum of both their radii, i.e.

$$\sqrt{(x_A - x_B)^2 + (Y_B - Y_B)^2 + (Z_A - Z_B)^2} \leq (R_A + R_B). \quad (A1)$$

Additionally, the distance between the surfaces of two spheres is calculated with

$$D_{AB} = \sqrt{(x_A - x_B)^2 + (Y_B - Y_B)^2 + (Z_A - Z_B)^2} - (R_A + R_B). \quad (A2)$$

In this research, the accurate Computer Aided Design (CAD) model of the manipulator was first converted into an oriented bounding box model as shown in Fig. A.1. Then, the oriented bounding box model was converted into a bounding sphere model with different levels of refinement (1, 8, 64, 512 or more spheres) as shown in Fig. A.2 for the RV-E2 manipulator. If the ensembles of spheres for different levels of refinement are hierarchically-organised, efficient and accurate collision detection can be achieved (Hubbard, 1996). Collision detection started at the refinement level at which objects were represented by a single sphere. If, at this highest level, no collision was detected between two spheres further refinement of the representation was not required. If a collision was detected, the algorithm proceeded with a more accurate representation using more spheres.

A similar procedure was followed to estimate the distance between two objects. The smallest distance between two

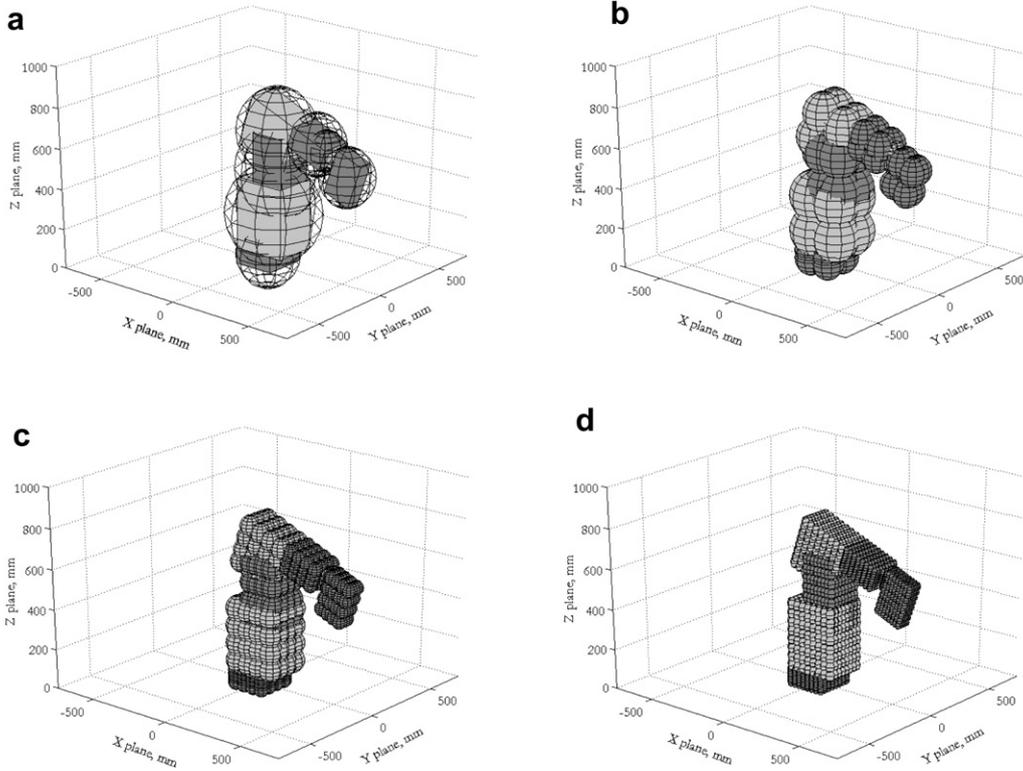


Fig. A.2 – Bounding sphere representations of the RV-E2 manipulator; (a) one sphere per link including the bounding box representation, (b) 8 spheres per link, (c) 64 sphere per link, (d) 512 spheres per link.

objects represented by an arbitrary number of spheres was estimated by calculating the distance between all the spheres representing both objects. The smallest distance found was used as an estimate of the smallest distance between both objects. Again, by using a higher level of refinement of the representation a higher level of accuracy of the estimate was obtained.

Appendix B. Partial analytic solution of the inverse kinematics of the seven-link manipulator

With the parameters of Table 1, the kinematic chain of the manipulator is fully defined up to the Q-point which is located at the intersection of the axes of joints 5, 6 and 7. The forward kinematics are governed by the equation

$${}^0_7T = {}^0_1T(d_1) \cdot {}^1_2T(\theta_2) \cdot {}^2_3T(\theta_3) \cdot {}^3_4T(\theta_4) \cdot {}^4_5T(\theta_5) \cdot {}^5_6T(\theta_6) \cdot {}^6_7T(\theta_7) \quad (B1)$$

where

$${}^{i-1}_i T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and 9_7T contains the translation and orientation of the Q-point of the manipulator with respect to the world co-ordinates as a function of the joint translation d_1 and the joint rotations θ_2 to θ_7 . For the cucumber-harvesting robot, the translation of the Q-point to the TCP is governed by the transformation matrix.

$${}^7_{TCP} T = \begin{bmatrix} 1 & 0 & 0 & 34 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 235 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (B2)$$

such that

$${}^0_{TCP} T = {}^7_{TCP} T \cdot {}^0_7T(d_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7). \quad (B3)$$

Because in the spherical wrist the last three axes intersect, for the RV-E2 6 link manipulator a closed form solution of the inverse kinematics can be obtained (Craig, 1989). With some modifications due to differences in the kinematic structure, the derivation of the inverse kinematics of the Mitsubishi RV-E2 is based on the derivation of the inverse kinematics of an Unimation PUMA 560 as described by Craig (1989).

The rotation of the first rotational axis θ_2 is obtained by rewriting Eq. (B1) as

$$[{}^0_2T(d_1, \theta_2)]^{-1} \cdot {}^0_7T = {}^2_3T(\theta_3) \cdot {}^3_4T(\theta_4) \cdot {}^4_5T(\theta_5) \cdot {}^5_6T(\theta_6) \cdot {}^6_7T(\theta_7) \quad (B4)$$

For ease of notation, hereafter $s_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$. Equating the (2,4) elements from both sides of Eq. (B4) results in

$$-s_2 p_x + c_2 (d_1 - p_z) = 0, \quad (B5)$$

from which follows that

$$\theta_2 = \text{Atan2}(d_1 - p_z, p_x) \quad (B6a)$$

and

$$\theta_2 = \text{Atan2}(d_1 - p_z, p_x) + \pi \quad (B6b)$$

with $\text{Atan2}(a,b)$ being the two-argument arctangent. With θ_2 obtained from Eq. (B6), the transform ${}^0_2T(d_1, \theta_2)$ is fully defined. The rotation θ_4 is obtained by equating the (1,4) and (3,4) elements from both sides of Eq. (B4):

$$c_2p_x + s_2(d_1 - p_z) = a_4c_3c_4 - d_5c_3s_4 - a_5s_3s_4 - d_5s_3c_4 + a_4c_3 + a_3 \quad (\text{B7})$$

and

$$\theta_3 + \theta_4 = \text{Atan2} \left(\frac{(a_4s_4 - d_5)(c_2p_x + s_2(d_1 - p_z) - a_3) - (a_4c_4 + a_5)(p_y - d_2)}{(a_4c_4 + a_5)(c_2p_x + s_2(d_1 - p_z) - a_3) + (a_4s_4 - d_5)(p_y - d_2)} \right). \quad (\text{B20})$$

$$p_y - d_2 = -a_5s_3c_4 + d_5s_3s_4 - a_5c_3s_4 - d_5c_3c_4 - a_4c_3. \quad (\text{B8})$$

Eqs. (B7) and (B8) are squared and the results are added to obtain

$$a_5c_4 - d_5s_4 = K \quad (\text{B9})$$

with

$$K = \frac{(c_2p_x + s_2(d_1 - p_z) - a_3)^2 + (p_y - d_2)^2 - a_4^2 - a_5^2 - d_5^2}{2a_4}. \quad (\text{B10})$$

Eq. (B9) can be solved by expressing a_5 and d_5 in polar coordinates:

$$d_5 = \rho \cdot \cos(\phi) \quad (\text{B11})$$

and

$$a_5 = \rho \cdot \sin(\phi). \quad (\text{B12})$$

Substitution into Eq. (B9) yields

$$c_4 \sin(\phi) - s_4 \cos(\phi) = \frac{K}{\rho} \quad (\text{B13})$$

from which it follows that:

$$\sin(\phi - \theta_4) = \frac{K}{\rho} \quad (\text{B14})$$

$$\theta_7 = \text{Atan2} \left(\begin{array}{l} (-c_2c_3c_4s_5 + c_2s_3s_4s_5 + s_2c_5)r_{11} - (-s_2c_3c_4s_5 + s_2s_3s_4s_5 - c_2c_5)r_{31} \\ + (s_3c_4s_5 + c_3s_4s_5)r_{21}, (-c_2c_3c_4s_5 + c_2s_3s_4s_5 + s_2c_5)r_{12} \\ - (-s_2c_3c_4s_5 + s_2s_3s_4s_5 - c_2c_5)r_{32} + (s_3c_4s_5 + c_3s_4s_5)r_{22} \end{array} \right). \quad (\text{B24})$$

and

$$\cos(\phi - \theta_4) = \pm \sqrt{1 - \frac{K^2}{\rho^2}}. \quad (\text{B15})$$

Then, from

$$(\phi - \theta_4) = \text{Atan2} \left(\frac{K}{\rho}, \pm \sqrt{1 - \frac{K^2}{\rho^2}} \right) \quad (\text{B16})$$

we obtain

$$\theta_4 = \text{Atan2}(a_5, d_5) - \text{Atan2} \left(K, \sqrt{a_5^2 + d_5^2 - K^2} \right) \quad (\text{B17})$$

and

$$\theta_4 = \text{Atan2}(a_5, d_5) - \text{Atan2} \left(K, -\sqrt{a_5^2 + d_5^2 - K^2} \right) \quad (\text{B18})$$

The rotation θ_3 is obtained from

$$\left[{}^0_4T(d_1, \theta_2, \theta_3, \theta_4) \right]^{-1} {}^0_7T = {}^4_5T(\theta_5) \cdot {}^5_6T(\theta_6) \cdot {}^6_7T(\theta_7). \quad (\text{B19})$$

Equating the (1,4) and (3,4) elements from both sides of Eq. (B19) yields after some algebraic manipulations

Since four combinations of solutions of θ_2 and θ_4 do exist, θ_3 will have four possible solutions as well. This means that a given wrist position can be achieved by four combinations of the three joint rotations θ_2 , θ_3 and θ_4 , as is illustrated by [Sciavicco and Siciliano \(1996, p. 64, Fig. 2.27\)](#).

Since the left hand side of Eq. (B19) is fully determined now, this equation can be used to solve for θ_5 by equating the (1,3) and (3,3) elements from both sides to obtain

$$\theta_5 = \text{Atan2}({}^4_7T_{3,3}, -{}^4_7T_{1,3}) \quad (\text{B21})$$

with ${}^4_7T_{a,b}$ being the (a,b) element of the transformation matrix 4_7T . This yields for θ_5 the equation:

$$\theta_5 = \text{Atan2}(-s_2r_{13} - c_2r_{33}, -(c_2c_3c_4 - c_2s_3s_4)r_{13} + (s_2c_3c_4 - s_2s_3s_4)r_{33} - (-s_3c_4 - c_3s_4)r_{23}). \quad (\text{B22})$$

This equation can only be used if $\theta_6 \neq 0$. This condition can be verified by checking whether

$$-s_2r_{13} - c_2r_{33} > 0 \quad (\text{B23a})$$

and

$$(c_2c_3c_4 - c_2s_3s_4)r_{13} - (s_2c_3c_4 - s_2s_3s_4)r_{33} + (-s_3c_4 - c_3s_4)r_{23} > 0. \quad (\text{B23b})$$

Then the rotation θ_7 is calculated with

If however $\theta_6 = 0$, the axes θ_5 and θ_7 line up and the total rotation depends on both θ_5 and θ_7 . In that case the (1,3) and (3,3) elements are approximately zero and the θ_5 and θ_7 can be obtained from

$$\theta_5 + \theta_7 = \text{Atan2}(-{}^4_7T_{3,1}, -{}^4_7T_{3,2}) = \text{Atan2}(s_2r_{11} + c_2r_{31}, s_2r_{12} + c_2r_{32}). \quad (\text{B25})$$

By setting $\theta_7 = 0$, Eq. (B25) can be solved for θ_5 . Then the rotation θ_6 is obtained from the following relation

$$\left[{}^0_5T(d_1, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5) \right]^{-1} \cdot {}^0_7T = {}^5_6T(\theta_6) \cdot {}^6_7T(\theta_7). \quad (\text{B26})$$

Since the left hand side is fully determined at this stage,

$$\theta_6 = \text{Atan2}(-{}^5_7T_{1,3}, {}^5_7T_{3,3}) \quad (\text{B27})$$

which yields

$$\theta_6 = \text{Atan2} \left(\begin{array}{l} -(c_2c_3c_4c_5 - c_2s_3s_4c_5 + s_2s_5)r_{13} + (s_2c_3c_4c_5 - s_2s_3s_4c_5 - c_2s_5)r_{33} \\ + (s_3c_4c_5 + c_3s_4c_5)r_{23}, -(c_2c_3s_4 + c_2s_3c_4)r_{13} \\ + (s_2c_3s_4 + s_2s_3c_4)r_{33} + (s_3s_4 - c_3c_4)r_{23} \end{array} \right) \quad (\text{B28})$$

For the rotations of the spherical wrist, two solutions do exist. The second solution is

$$\begin{aligned}\theta'_5 &= \theta_5 + \pi \\ \theta'_6 &= -\theta_6 \\ \theta'_7 &= \theta_7 + \pi\end{aligned}\quad (\text{B29})$$

REFERENCES

- Chapelle, F., & Bidaud, P. (2004). Closed form solutions for inverse kinematics approximation of general 6R manipulators. *Mechanism and Machine Theory*, 39, 323–338.
- Craig, J. J. (1989). *Introduction to robotics*. Reading, MA, USA: Addison-Wesley.
- Hubbard, P. M. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15, 179–210.
- Jiménez, P., Thomas, F., & Torras, C. (2001). 3D collision detection: a survey. *Computational Graphics*, 25, 269–285.
- Nearchou, A. C. (1998). Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. *Journal of Mechanics and Machine Theory*, 33, 273–292.
- Parker, J. K., Khoogar, A. R., & Goldberg, D. E. (1989). Inverse kinematics of redundant robots using genetic algorithms. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1, 271–276.
- Sciavicco, L., & Siciliano, B. (1996). *Modeling and control of robot manipulators*. New York: McGraw-Hill.
- Van Henten, E. J., Hemming, J., Van Tuijl, B. A. J., Kornet, J. G., & Bontsema, J. (2003). Collision-free motion planning for a cucumber picking robot. *Biosystems Engineering*, 86, 135–144.
- Van Henten, E. J., Hemming, J., Van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., et al. (2002). An autonomous robot for automated harvesting of vegetable fruit 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. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering*, 86, 305–313.
- Van Henten, E. J., Van Tuijl, B. A. J., Hoogakker, G. J., Van der Weerd, M. J., Hemming, J., Kornet, J. G., et al. (2006). An autonomous robot for de-leafing cucumber plants grown in a high wire cultivation system. *Biosystems Engineering*, 94, 317–323.
- Van Henten, E. J., Van Willigenburg, L. G., & Van't Slot, D. (2009). Optimal design of a cucumber harvesting robot. *Computers and Electronics in Agriculture*, 65, 247–257.
- Van Willigenburg, L. G., Hol, C. W. J., & Van Henten, E. J. (2004). On-line near minimum-time path planning and control of an industrial robot picking fruits. *Computers and Electronics in Agriculture*, 44, 223–237.
- Zhao, J., & Badler, N. I. (1994). Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13, 313–336.