
Analytical solution of a five-degree-of-freedom inverse kinematics problem for the handling mechanism of an agricultural robot

Sinem Gozde Defterli* and Yunjun Xu

Mechanical and Aerospace Engineering Department,
University of Central Florida,
4000 Central Florida Blvd,
Orlando, FL 32816, USA
Email: sgdefterli@knights.ucf.edu
*Corresponding author

Abstract: Computing joint commands in real time by solving an inverse kinematic (IK) problem is an important manipulation task that is commonly encountered in agricultural operations such as fruit harvesting, disease detection and leaf sampling. To date, different numerical and analytical approaches have been proposed, and many of them are able to take constraints into account. In this study, a set of analytical algorithms is investigated to quickly solve the constrained five-degree-of-freedom inverse kinematic problem. The constraints due to actuator limitations and mechanical design of the mechanism are utilised in generating the solution sets for joint variables. Also a suboptimal solution for the inverse kinematic problem within this set is derived. The proposed methods are compared with a typical numerical solution based on non-linear constrained optimisation. It is shown that the proposed analytical and suboptimal semi-analytical solutions can be found in a much quicker fashion as compared to the non-linear constrained optimisation approach. The algorithms are validated by the handling mechanism in a ground robot platform specifically designed to operate in strawberry orchards.

Keywords: agricultural robotics; analytical modelling; inverse kinematics; numerical optimisation; robotic manipulators.

Reference to this paper should be made as follows: Defterli, S.G. and Xu, Y. (2017) ‘Analytical solution of a five-degree-of-freedom inverse kinematics problem for the handling mechanism of an agricultural robot’, *Int. J. Mechanisms and Robotic Systems*, Vol. 4, No. 1, pp.43–60.

Biographical notes: Sinem Gozde Defterli received the B.S. and M.S. degrees from the Department of Mechanical Engineering in Middle East Technical University, Ankara, Turkey in 2010 and 2014, respectively. She got a Minor degree in “Engineering Materials and Alloys” from the Department of Metallurgical and Material Engineering in Middle East Technical University in 2011. Currently, she is a PhD student of Autonomous Robotics and Control Laboratory in Mechanical and Aerospace Engineering Department at the University of Central Florida. Her research interests include robotics, control, kinematics and dynamics.

Yunjun Xu received the B.S. and M.S. degrees from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1996 and 1999, respectively. Also he received the M.S. and PhD degrees from the University of Florida, Gainesville, FL, in 2002 and 2003, respectively. Currently, he is an Associate

Professor in Department of Mechanical and Aerospace Engineering at the University of Central Florida. His research interests include linear and non-linear control theory and application, orbital and attitude dynamics modelling and simulation, robotics, navigation, guidance and control system and virtual environments.

1 Introduction

There is a growing interest in applying robotic and automation technologies to agriculture fields to reduce labour cost and enhance product quality (Schmoltdt, 2012). Different types of robots have been developed to help or replace labour work in operations such as harvesting, yield prediction and weed control (Hajjaj and Sahari, 2014; Rodriguez et al., 2013; Cui et al., 2013; Defterli et al., 2016; Misaghi et al., 2004; Slaughter et al., 2008). Along with emerging ground and aerial agricultural robots, associated technologies have also advanced in the areas such as computer vision (Milella et al., 2006), guidance and control (Li et al., 2015; Wang, 2010), spectrum analyses (Wang et al., 2012), etc. Inverse kinematics (IK), an important technique in agriculture automation, is widely practiced at agricultural operations in orchards, e.g. fruit picking in harvesting robots (Van Henten et al., 2010).

An inverse kinematics problem is to find the joint commands for the end-effector to reach a desired position in the workspace of a robot. Such a problem is challenging for redundant robot manipulators, in which the number of joint variables is larger than the number of equations, considering computational cost, optimality and robustness. Many methods have been proposed to address inverse kinematic problems (Aristidou and Lasenby, 2009; Baerlocher, 2001), which can be broadly categorised into analytical methods, numerical methods and mesh-based methods.

Analytical solutions can be found for some manipulators (Gan et al., 2005; Ozgoren, 2013; Shimizu et al., 2008; Xu and She et al., 2014; Tolani et al., 2000). For example in Gan et al. (2005), an analytical solution is derived for a five-degree-of-freedom (DOF) manipulator via the Denavit-Hartenberg method. In Ozgoren (2013), two types of inverse kinematic problems, addressing position and velocity, respectively, have been solved analytically. An analytical approach is proposed for solving the inverse kinematic problem of a seven-DOF redundant manipulator (Shimizu et al., 2008), in which joint constraints are considered. In general, analytical approaches have less computational cost and are reliable; however only simple manipulators could have analytical solutions, most of which are not optimal (Waldron and Schimiedeler, 2008).

Numerical methods are developed for solving inverse kinematic problems in most manipulators (Chirikjian, 1993). First, the Jacobian-based method (David and Schrader, 1984) and its variations such as the Jacobian transpose (Maciejewski and Klein, 1985), pseudo-inverse (Balestrino et al., 1984), damped least squares (Nakamura and Hanafusa, 1986) and selectively damped least squares (Buss and Kim, 2005) have been extensively studied. However, sometimes the Jacobian-based methods have singularity issues. Second, inverse kinematic problems have been formulated as a feedback control problem, and the Lyapunov theorem has been used to prove stability (Sciavicco and Siciliano,

1988; Colome and Torras, 2015; Novakovic and Nemeč, 1990; Falco and Natale, 2011), as in Novakovic and Nemeč (1990), where a sliding mode control method is proposed. It is claimed that no Jacobian matrix is needed and therefore singularities are avoided. Third, a more general approach in solving inverse kinematic problems, particularly constrained ones, is to formulate them as non-linear constrained optimisation problems (Chirikjian, 1993; Jin et al., 2006; Mitsi et al., 1995; Manocha and Canny, 1994). Additionally, evolutionary algorithms (Tabandeh et al., 2006; Nearchou, 1998) and neural network methods (Bingul et al., 2005; Duka, 2014; Hasan et al., 2011) have been used in solving inverse kinematic problems. Numerical approaches can be applied to any kind of inverse kinematic problems; however, in general the computational cost is high and whether or not a solution can be found is not guaranteed.

There are also a few papers investigating mesh-based strategies for solving inverse kinematic problems (Sumner et al., 2005; Tarokh and Kim, 2007). For example, in Tarokh and Kim (2007), inverse solutions for different decomposed cells are tabulated, and the joint command is directly read from the table based on the corresponding index. This approach requires a prior knowledge about the workspace and a large number of solutions (that can be interpolated) available beforehand.

In this paper, the constrained 5-DOF inverse kinematic problem of a handling mechanism is studied for an agricultural robot specifically designed for strawberry fields (Xu and Ehsani et al., 2014). The mechanism consists of a 3-DOF XYZ table and a 3-DOF robotic arm that is serially connected by revolute joints. Typically a strawberry plant in commercial fields has a sphere like shape when it is fully grown. In inverse kinematics calculations, this characteristic of the plant aids to set certain bound values for the locations of diseased leaves in an approximated sphere volume.

The advantages of the proposed methods, as compared with a typical non-linear constrained optimisation solution (NCOS), are summarised here:

- 1 The computational cost of the new methods is small and can be used in real time
- 2 An optimal solution within the defined solution set is found.

For a typical NCOS, local minima are often found depending on the quality of the initial guess. Therefore, the solution found in proposed methods can be used as the initial guess for the NCOS to converge quicker.

The paper is organised as follows. First, the kinematic relationships (i.e. mechanism constants, joint variables and body reference frames) are described for the handling mechanism of an agricultural ground robot (AGR) in Section 2 and the forward kinematic equations are discussed. For the end-effector to reach a certain 3D coordinate, the joint variables of the handling mechanism are found by utilising different methods. To this purpose, the analytical and semi-analytical suboptimal solution methods are derived based on the mechanism constraints and geometric information in Section 3. Then simulation and experiment results are shown in Sections 4 and 5, and their performances are compared with that of the NCOS. Conclusions are given in Section 6.

2 Problem definition

The agricultural-robot-handling mechanism is composed of a 3-DOF XYZ table and a 3-DOF manipulator arm as illustrated in Figure 1. The three translational variables of the XYZ table are denoted by s_{10} , s_{21} and s_{32} . The three rotational variables of the manipulator

has no effect on the position of tip point; however, it will affect the alignment of the end-effector (containing a cutter and a holder) and how the tip point touches the diseased leaf. Thus, the inverse kinematic problem modelled here is a 5-DOF problem.

The inverse kinematic problem is defined as: solve for $\{s_{10}, s_{21}, s_{32}, \theta, \beta\}$ such that the tip point position $[x_{P/O}, y_{P/O}, z_{P/O}]^T$ calculated using (2)–(4) matches the target diseased leaf position $T(x_T, y_T, z_T)$, considering the following constraints:

$$s_{10} \in [s_{10,\min}, s_{10,\max}], \quad s_{21} \in [s_{21,\min}, s_{21,\max}], \quad s_{32} \in [s_{32,\min}, s_{32,\max}], \quad \beta \in [\beta_{\min}, \beta_{\max}] \quad \text{and} \\ \theta \in [\theta_{\min}, \theta_{\max}].$$

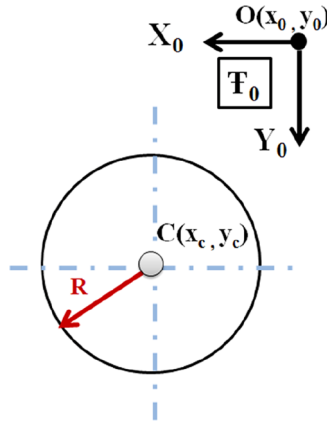
3 Analytical and suboptimal semi-analytical solutions

In this section, a set of analytical solutions (AS) is derived for the inverse kinematic problem defined in Section 2. Within this set, a suboptimal semi-analytical solution (SSAS) is further derived. Additionally, a non-linear constrained optimisation algorithm, widely used in solving inverse kinematic problems, is presented for comparison.

3.1 Analytical solution

The inverse kinematic problem is solved for the following scenario. The volume of a fully grown plant is assumed to be encircled by a sphere. Its top view is shown with a radius of R and a centre point $C(x_C, y_C)$ in Figure 2.

Figure 2 Circular area covered by a strawberry plant (see online version for colours)



By inserting the minimum and maximum values of the joint variables $\{s_{10}, s_{21}, s_{32}, \theta, \beta\}$ into the forward kinematics (2)–(4), the target point $T(x_T, y_T, z_T)$ in frame F_0 should be within the limits calculated by

$$x_{\min} = s_{21,\min} + d_x - (l_2 + l_3 \cos \beta_{\min}) \sin \theta_{\max} \quad (5)$$

$$x_{\max} = s_{21,\max} + d_x - (l_2 + l_3 \cos \beta_{\max}) \sin \theta_{\min} \quad (6)$$

$$y_{\min} = s_{10,\min} + l_1 - d_y + l_3 \sin \beta_{\min} \quad (7)$$

$$y_{\max} = s_{10,\max} + l_1 - d_y + l_3 \sin \beta_{\max} \quad (8)$$

$$z_{\min} = -s_{32,\min} + d_z - (l_2 + l_3 \cos \beta_{\min}) \cos \theta_{\min} \quad (9)$$

and

$$z_{\max} = -s_{32,\max} + d_z - (l_2 + l_3 \cos \beta_{\max}) \cos \theta_{\max} \quad (10)$$

There is no solution if the target point $T(x_T, y_T, z_T)$ is not within the ranges as defined in (5)–(10). In this case, the ground robot should adjust its translation and orientation, so that the target diseased leaf can be in the range.

Based on (3), target coordinates $T(x_T, y_T, z_T)$ and the range of $\beta \in [\beta_{\min}, \beta_{\max}]$, the first set for s_{10} is limited by the set $S_{s_{10}}^{(1)}$ as

$$S_{s_{10}}^{(1)} : [y_T - l_3 \sin \beta_{\max} - l_1 + d_y, y_T - l_3 \sin \beta_{\min} - l_1 + d_y] \quad (11)$$

In addition, s_{10} should satisfy its upper and lower bounds, which is the second set $S_{s_{10}}^{(2)} : [s_{10,\min}, s_{10,\max}]$. Therefore, s_{10}^* should be selected from the intersection set $S_{s_{10}}$ as

$$S_{s_{10}} = S_{s_{10}}^{(1)} \cap S_{s_{10}}^{(2)} \quad (12)$$

Once $s_{10}^* \in S_{s_{10}}$ is selected, β^* is calculated based on (3) as

$$\beta^* = \sin^{-1} \left[(y_T - s_{10}^* - l_1 + d_y) / l_3 \right] \quad (13)$$

Since the range of β is already considered in generating the set for s_{10}^* , β^* calculated via (13) is already within its limit.

Based on (2), the value of θ is limited by the first set $S_{\theta}^{(1)}$ as

$$S_{\theta}^{(1)} : \left[\sin^{-1} \left(\frac{x_T - s_{21,\min} - d_x}{-l_2 - l_3 \cos \beta^*} \right), \sin^{-1} \left(\frac{x_T - s_{21,\max} - d_x}{-l_2 - l_3 \cos \beta^*} \right) \right] \quad (14)$$

In addition to that, based on (4), the value of θ is limited by the second set $S_{\theta}^{(2)}$ as

$$S_{\theta}^{(2)} : \left[\cos^{-1} \left(\frac{z_T + s_{32,\min} - d_z}{-l_2 - l_3 \cos \beta^*} \right), \cos^{-1} \left(\frac{z_T + s_{32,\max} - d_z}{-l_2 - l_3 \cos \beta^*} \right) \right] \quad (15)$$

Furthermore, θ is constrained by its motion capability due to the power restrictions of the motors $\theta \in S_{\theta}^{(3)} : [\theta_{\min}, \theta_{\max}]$. Therefore, θ^* is chosen from the intersection set $\theta^* \in S_{\theta}$ formed by

$$S_{\theta} = S_{\theta}^{(1)} \cap S_{\theta}^{(2)} \cap S_{\theta}^{(3)} \quad (16)$$

After s_{10}^* , β^* and θ^* are selected and inserted in (2) and (4), s_{21}^* and s_{32}^* are then calculated by

$$s_{21}^* = x_T - d_x + (l_2 + l_3 \cos \beta^*) \sin \theta^* \quad (17)$$

and

$$s_{32}^* = -z_T - d_z - (l_2 + l_3 \cos \beta^*) \cos \theta^* \quad (18)$$

respectively.

Remark 1: There are many solutions depending on the selection of s_{10}^* . One approach is to select the middle point of the set $S_{s_{10}}$ as s_{10}^* .

Remark 2: There are also many solutions in this algorithm depending on the selection of θ^* within set S_θ . For example, the minimum absolute value of the intersection set S_θ can be selected as θ^* to maximise the value of $\cos \theta^*$. A larger value of $\cos \theta^*$ in (18) will increase the possibility that the calculated s_{32}^* does not exceed its geometric limits while reaching the target coordinate z_T .

The algorithm for finding the analytical solutions is listed in Table 1 as Algorithm 1. It is worth noting that

- 1 the mechanism always goes to its reset position and orientation (i.e. the original pose) after a picking task is finished
- 2 the motion sequence of the handling mechanism to reach the target coordinates must be performed to avoid collision between the arm and the plant canopy, as in the following remark.

Table 1 Algorithm 1 analytical solution

Step 0:	Get the target coordinate.
Step 1:	Check whether the target point is inside of the solution limits using (5)–(10). If this condition is not satisfied, there is no inverse kinematics solution.
Step 2:	Create the intersection set $S_{s_{10}}$ for s_{10}^* using (12).
Step 3:	Compute β^* value using (13).
Step 4:	Create the intersection set S_θ for θ^* using (16).
Step 5:	Calculate the s_{21}^* and s_{32}^* values using (17) and (18).
Step 6:	Implement the joint commands following the Command Sequence
Step 7:	Once the task is accomplished, the arm will go back to its reset position (i.e. $s_{10} = s_{10,\min}$, $s_{21} = s_{21,\min}$, $s_{32} = s_{32,\min}$, $\theta = 0^\circ$ and $\beta = 0^\circ$).

Remark 3: (The Command Sequence): Open the end-effector, translate along the z -axis by $s_{32}^* + h_{\text{extra}}$, translate along the y -axis by s_{10}^* , translate along the x -axis by s_{21}^* , rotate about the y -axis by θ^* , rotate about the x -axis by β^* , rotate about z -axis by ψ^* and,

lastly, translate along the z -axis by h_{extra} to end at s_{32}^* . After the commands are executed the end-effector is closed. Please note that h_{extra} is added in the z -axis translation at the beginning, so the arm can get a larger free space at the top of plants to prevent touching the canopy while adjusting its pose.

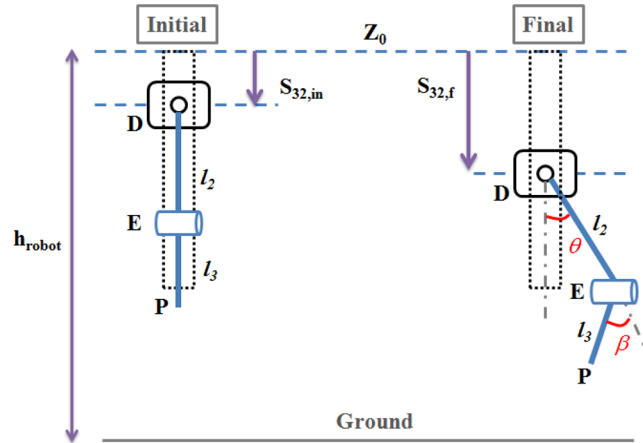
3.2 Suboptimal semi-analytical solution

In the previous section, a set of analytical algorithms is studied and a specific inverse kinematics solution in this set is derived, which depends on how s_{10}^* and θ^* are selected.

In this section, the minimum energy solution in the algorithm set is derived. The performance index J_{total} is defined as $J_{\text{total}} = J_{\text{pot}} + J_{\text{fric}}$, where J_{total} , J_{pot} and J_{fric} are the total energy required, the potential energy change and the frictional energy loss, respectively.

Figure 3 illustrates how the potential energy changes before and after the manipulator arm accomplishes the translational and rotational commands. $s_{32,\text{in}}$ and $s_{32,\text{f}}$ are the initial and the final translations along the z -axis, respectively.

Figure 3 Posture change of the arm when it reaches a certain target (see online version for colours)



Based on Figure 3, the initial potential energy of the arm is

$$PE_{\text{in}} = m_2 g (h_{\text{robot}} - s_{32,\text{in}} - l_2/2) + m_3 g (h_{\text{robot}} - s_{32,\text{in}} - l_2 - l_3/2) \quad (19)$$

and the final potential energy is

$$PE_{\text{f}} = m_2 g (h_{\text{robot}} - s_{32,\text{f}} - 0.5l_2 \cos \theta) + m_3 g (h_{\text{robot}} - s_{32,\text{f}} - (l_2 + 0.5l_3 \cos \beta) \cos \theta) \quad (20)$$

where m_2 and m_3 are the masses of the links DE and $EF + FP$, respectively. Then, the potential energy change is

$$J_{\text{pot}} = PE_{\text{f}} - PE_{\text{in}} = (m_2 + m_3) g (s_{32,\text{in}} - s_{32,\text{f}}) + (m_2/2 + m_3) g l_2 (1 - \cos \theta) + m_3 g l_2 / (2(1 - \cos \beta \cos \theta)) \quad (21)$$

The friction on the revolute joints of the manipulator arm is neglected and the friction loss due to the translational motion of the XYZ table is calculated as

$$J_{\text{fric}} = \mu W \left[(s_{10,f} - s_{10,\text{in}}) + (s_{21,f} - s_{21,\text{in}}) + (s_{32,f} - s_{32,\text{in}}) \right] \quad (22)$$

Here μ is the friction coefficient between the sliders and the sliding axes along the x , y and z axes. W is the total resultant weight of the mechanism. The initial position of the sliders are represented as $s_{10,\text{in}}$, $s_{21,\text{in}}$ and $s_{32,\text{in}}$, respectively. When the mechanism is commanded to reach the target coordinate, it moves as $s_{10,f} \geq s_{10,\text{in}}$, $s_{21,f} \geq s_{21,\text{in}}$ and $s_{32,f} \geq s_{32,\text{in}}$. Thus, $J_{\text{fric}} \geq 0$.

The minimum energy solution happens when

$$\frac{\partial J_{\text{total}}}{\partial s_{10}} = \frac{\partial J_{\text{pot}}}{\partial s_{10}} + \frac{\partial J_{\text{fric}}}{\partial s_{10}} = 0 \quad (23)$$

where the partial derivatives $\frac{\partial J_{\text{pot}}}{\partial s_{10}}$ and $\frac{\partial J_{\text{fric}}}{\partial s_{10}}$ are derived as

$$\begin{aligned} \frac{\partial J_{\text{pot}}}{\partial s_{10}} = & - \left[m_2 (l_2/2 + l_3 \cos \beta) g \sin \theta \right] \frac{\partial \theta}{\partial \beta} \frac{\partial \beta}{\partial s_{10}} - \left[(m_3 l_3 g \cos \beta \sin \theta) / 2 \right] \frac{\partial \theta}{\partial \beta} \frac{\partial \beta}{\partial s_{10}} \\ & + \left[(-m_2 - m_3 / 2) l_3 g \sin \beta \cos \theta \right] \frac{\partial \beta}{\partial s_{10}} \end{aligned} \quad (24)$$

and

$$\begin{aligned} \frac{\partial J_{\text{fric}}}{\partial s_{10}} = & \left[\mu W (l_2 + l_3 \cos \beta) (\cos \theta + \sin \theta) \right] \frac{\partial \theta}{\partial \beta} \frac{\partial \beta}{\partial s_{10}} \\ & + \left[\mu W l_3 \sin \beta (\cos \theta - \sin \theta) \right] \frac{\partial \beta}{\partial s_{10}} + [\mu W] \end{aligned} \quad (25)$$

where $\frac{\partial \beta}{\partial s_{10}}$ can be derived from (3) as

$$\frac{\partial \beta}{\partial s_{10}} = \left[-1 / \sqrt{l_3^2 - (y_T - s_{10} - l_2 + d_y)^2} \right] \quad (26)$$

There are two possible solutions of θ depending on which of the two sets $S_{\theta}^{(1)}$ and $S_{\theta}^{(2)}$ has the higher minimum bound. Therefore the calculation of $\partial \theta / \partial \beta$ should also consider these two cases.

Case 1: For set $S_{\theta}^{(1)}$, $\partial \theta / \partial \beta$ is derived as

$$\frac{\partial \theta}{\partial \beta} = \frac{\partial}{\partial \beta} \left[\sin^{-1} \left(\frac{x_T - s_{21,\text{min}} - d_x}{-l_2 - l_3 \cos \beta} \right) \right] = \frac{\partial g_1}{\partial h_1} \frac{\partial h_1}{\partial f} \frac{\partial f}{\partial \beta} \quad (27)$$

in which $f(\beta) \triangleq \cos \beta$, $h_1(f) \triangleq (x_T - s_{21,\text{min}} - d_x) / (-l_2 - l_3 f)$ and $g_1(h_1) \triangleq \sin^{-1}(h_1)$.

Case 2: For set $S_\theta^{(2)}$, $\partial\theta/\partial\beta$ is derived as

$$\frac{\partial\theta}{\partial\beta} = \frac{\partial}{\partial\beta} \left[\cos^{-1} \left(\frac{z_T + s_{32,\min} - d_z}{-l_2 - l_3 \cos\beta} \right) \right] = \frac{\partial g_2}{\partial h_2} \frac{\partial h_2}{\partial f} \frac{\partial f}{\partial\beta} \quad (28)$$

in which $h_2(f) \triangleq (z_T + s_{32,\min} - d_z)/(-l_2 - l_3 f)$ and $g_2(h_2) \triangleq \cos^{-1}(h_2)$.

After comparing the values of $\partial\theta/\partial\beta$ in (27) and (28), the lowest value of $\partial\theta/\partial\beta$ is chosen to be implemented.

P1: A non-linear constrained optimisation problem is formulated to be solved for s_{10}^* in order to achieve $\partial J_{total}/\partial s_{10}|_{s_{10}=s_{10}^*} = 0$. The “fmincon” function in MATLAB[®] is used as the numerical solver. The performance index is

$$J = |\partial J_{total}/\partial s_{10}| \quad (29)$$

which is subjected to the boundary constraint of $s_{10}^* \in [s_{10,\min}, s_{10,\max}]$. In addition, the following inequality and equality constraints are implemented to avoid any singularities that may arise from the \sin^{-1} , \cos^{-1} and square root functions in (14), (15) and (26). The conditions

$$\left[(l_2 + l_3 \cos\beta)(\cos\theta + \sin\theta) \right] \frac{\partial\theta}{\partial\beta} \frac{\partial\beta}{\partial s_{10}} + \left[l_3 \sin\beta(\cos\theta - \sin\theta) \right] \frac{\partial\beta}{\partial s_{10}} + 1 = 0 \quad (30)$$

$$y_T - s_{10} - l_2 + d_y - l_3 < 0 \quad (31)$$

$$-y_T + s_{10} + l_2 - d_y - l_3 < 0 \quad (32)$$

$$-x_T + s_{21,\min} + d_x - l_2 - l_3 \cos\beta < 0 \quad (33)$$

$$x_T - s_{21,\min} - d_x - l_2 - l_3 \cos\beta < 0 \quad (34)$$

$$z_T + s_{32,\min} - d_z - l_2 - l_3 \cos\beta < 0 \quad (35)$$

and

$$-z_T - s_{32,\min} + d_z - l_2 - l_3 \cos\beta < 0 \quad (36)$$

Here, β , θ , $\partial\theta/\partial\beta$ and $\partial\beta/\partial s_{10}$ are functions of s_{10} .

The semi-analytical suboptimal algorithm is shown in Table 2 as Algorithm 2.

Table 2 Algorithm 2 Suboptimal semi-analytical solution

Step 0:	Get the target coordinates.
Step 1:	Check whether or not the target point is inside of the solution limits using (5)–(10). If this condition is not satisfied, there is no solution.
Step 2:	Solve problem P1 for s_{10}^* satisfying all the constraints defined in (30)–(36) value. The following initial conditions are set: $s_{10} = s_{10,\min}$, $s_{21} = s_{21,\min}$, $s_{32} = s_{32,\min}$, $\theta = 0^\circ$ and $\beta = 0^\circ$.
Step 3:	Continue with Steps 3–7 in Algorithm 1.

Remark 4: The minimisation problem **P1** is solved by considering the variation of s_{10} only. However, as mentioned in Remark 2, a specific solution within the algorithm set is also dependent on how θ is chosen in its set S_θ . In the minimum energy solution proposed here, the θ^* value in the set S_θ is selected as mentioned in Remark 2. Therefore the solution is suboptimal.

Remark 5: In addition to the analytical solution steps, a numerical scheme is used to solve **P1**. Therefore the solution method is semi-analytical.

3.3 Non-linear constrained optimisation solution

A NCOS is briefly listed as a typical numerical approach for solving the inverse kinematic problem. The performance of the NCOS is compared with the algorithms proposed in this study.

P2: The objective function here is stated in (37). The “fmincon” function in MATLAB[®] is used as the solver. $s_{10} = s_{10,\min}$, $s_{21} = s_{21,\min}$, $s_{32} = s_{32,\min}$, $\theta = 0^\circ$ and $\beta = 0^\circ$ are used as the initial guesses.

$$J = J_{pot} + J_{fric} \quad (37)$$

In **P2**, the minimum energy problem is subjected to the non-linear equality constraints in order to satisfy the forward kinematics equations stated in (2)–(4). The inequality constraints are due to the upper and the lower bounds of each joint variable and are given as $s_{10}^* \in [s_{10,\min}, s_{10,\max}]$, $s_{21}^* \in [s_{21,\min}, s_{21,\max}]$, $s_{32}^* \in [s_{32,\min}, s_{32,\max}]$, $\theta^* \in [\theta_{\min}, \theta_{\max}]$ and $\beta^* \in [\beta_{\min}, \beta_{\max}]$.

Remark 6: Although a numerical scheme is used to solve $\partial J_{total} / \partial s_{10} |_{s_{10}=s_{10}^*} = 0$ in the SSAS, its computation cost is still smaller than that of the NCOS, as to be shown in the simulation and experiment sections.

4 Simulations

All three algorithms are run on the same computer with a 64-bit operating system, an Intel[®] Core[™] i7 920 processor (2.67 GHz) and 6GB of RAM of using a 64bit operating system. All algorithms are executed in the MATLAB[®] R2009a.

Once the target point is entered into the AS, the SSAS and the NCOS algorithms, the joint variables are calculated, and the CPU time and their performance indices are recorded, as shown in Table 3.

Table 3 Performance indices and average CPU time of three methods

Target Point in F_0 [mm]			Performance Index			CPU Time (s)			Performance Index Increasing w.r.t NCOS (%)		CPU Time Reduction w.r.t NCOS (%)	
									SSAS	AS	SSAS	AS
x	y	z	NCOS	SSAS	AS	NCOS	SSAS	AS	SSAS	AS	SSAS	AS
426	345	-331.6	5950.5	6326.9	6377.3	0.576	0.300	0.014	6.33	7.17	47.92	97.57
450	285	-368.6	6505.9	6882.3	6918.3	0.549	0.306	0.015	5.79	6.34	44.26	97.27
489	270	-375.6	6913.8	7290.3	7299.2	0.605	0.294	0.015	5.45	5.57	51.40	97.52
445	325	-323.1	5732.0	6108.4	6158.8	0.561	0.300	0.015	6.57	7.45	46.52	97.33
410	350	-355.1	6418.0	6794.4	6844.8	0.552	0.296	0.015	5.86	6.65	46.38	97.28
520	350	-298.6	6111.9	6511.5	6538.7	0.593	0.295	0.014	6.54	6.98	50.25	97.64
568	310	-349.5	7438.9	7838.6	7865.7	0.602	0.299	0.015	5.37	5.74	50.33	97.51
520	280	-350.9	6708.2	7107.9	7110.2	0.561	0.315	0.015	5.96	5.99	43.85	97.33
580	310	-377.9	8252.1	8651.6	8678.9	0.564	0.298	0.014	4.84	5.17	47.16	97.52
525	280	-353.5	6821.8	7221.4	7223.8	0.544	0.302	0.015	5.86	5.89	44.49	97.24
535	450	-350.6	8516.2	8909.0	8943.0	0.585	0.297	0.016	4.61	5.01	49.23	97.26
510	450	-339.4	7995.0	8387.9	8421.8	0.568	0.300	0.016	4.91	5.34	47.18	97.18
565	420	-340.1	8256.9	8649.8	8683.7	0.599	0.297	0.015	4.76	5.17	50.42	97.50
536	460	-382.8	9414.1	9807.0	9841.0	0.543	0.304	0.015	4.17	4.53	44.01	97.24
580	427	-387.3	9630.6	10023.5	10057.4	0.529	0.301	0.018	4.08	4.43	43.10	96.60
450	400	-314.7	6311.3	6681.3	6738.1	0.589	0.307	0.016	5.86	6.76	47.88	97.28
410	410	-385.3	7748.2	8118.2	8175.1	0.554	0.298	0.015	4.78	5.51	46.21	97.29
475	390	-300.8	6117.5	6487.5	6544.4	0.565	0.304	0.014	6.05	6.98	46.19	97.52
435	440	-372.2	7966.7	8336.6	8393.5	0.574	0.296	0.015	4.64	5.36	48.43	97.39
482	465	-381.9	8912.0	9282.0	9338.9	0.540	0.304	0.015	4.15	4.79	43.70	97.22

As expected, the proposed AS algorithm has the lowest CPU time with average value of 15.1 ms (standard deviation 0.912 ms), while the NCOS algorithm takes the longest time with average value of 567.7 ms (standard deviation 22.2 ms) to reach a solution while satisfying the constraints. It is worth mentioning that, depending on the initial guess, a solution is not guaranteed when the NCOS is applied. The performance index of the NCOS algorithm has the lowest values, because it solves for the minimum energy solution within the search space. In the SSAS algorithm, the resultant performance index is higher than that of the NCOS but lower than that of the AS, as expected. As mentioned in Remark 4, the SSAS only searches for the minimum solution in a subspace defined by s_{10} .

The comparison of the AS and SSAS algorithms with respect to the NCOS are also listed in Table 3. For example, the change in performance indices is calculated for the SSAS with respect to the NCOS using $(J_{SSAS} - J_{NCOS})100/J_{NCOS}$ and the amount of change in CPU time is calculated by $|(t_{SSAS} - t_{NCOS})100/t_{NCOS}|$. As seen from Table 3, the performance indices of the AS and the SSAS are not significantly increased as compared

to those of the NCOS. However, the CPU time reduction is large in the AS and SSAS, as compared to those in the NCOS.

The resultant poses of the handling mechanism for the same target point $[x_T, y_T, z_T] = [426, 345, -331.61]$ mm are animated in the MATLAB® environment for all three methods as shown in Figures 4–6.

Figure 4 Pose of the handling mechanism solved by the AS (see online version for colours)

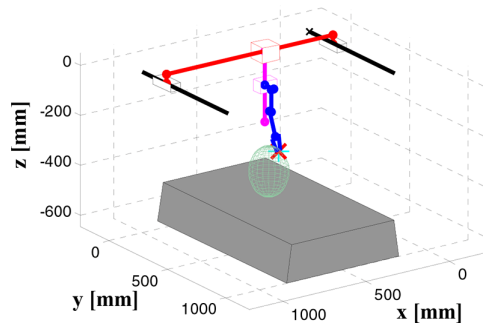


Figure 5 Pose of the handling mechanism solved by the SSAS (see online version for colours)

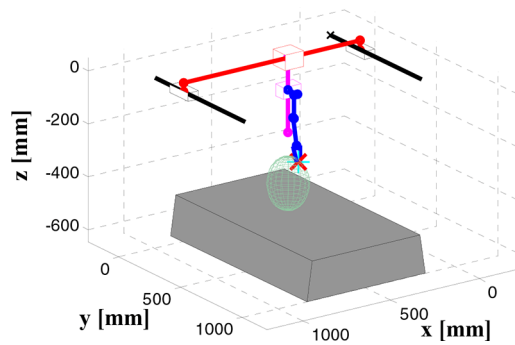
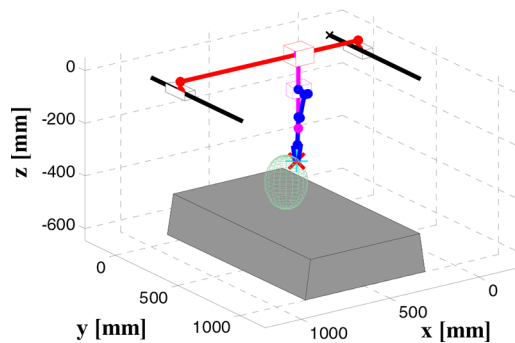


Figure 6 Pose of the handling mechanism solved by the NCOS (see online version for colours)



As seen from the Figures 4–6, the handling mechanism is shown and the grey trapezoidal volume represents a plant bed that the robot travels over a strawberry bed. The green spherical volume corresponds to a fully grown plant. All the target points to be reached are assumed to be on the surface of this spherical volume.

5 Hardware set-up and experiments

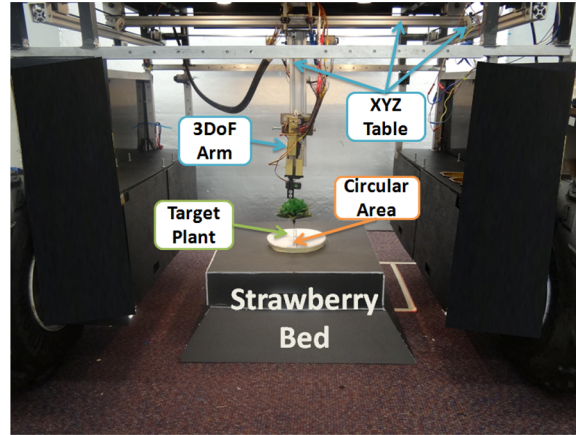
5.1 Hardware configuration in experimental set-up

The proposed algorithms are tested in a handling platform on-board of a mobile robot. The platform provides the execution of the translational and rotational motions on x , y and z axes. The pulley-belt-based translational motion is constructed by eight pulleys which are XL series lightweight pulleys with an outer diameter of 4.14 cm, 22 teeth and a maximum width of 0.95 cm. The belt is a trapezoidal tooth neoprene timing belt that has an XL trade size with a 0.51-cm pitch and a 0.95-cm width. The rotational motion is realised using a manipulator arm whose links are serially connected with revolute joints via the Lynx motion Little Grip Kit and Hi-Tech HS 422 servo motors. Additionally, the end-effector has an open/close function via two fingers for picking and holding the target leaf. The kinematic constraints in the platform are, $75\text{mm} \leq s_{10} \leq 660\text{mm}$, $80\text{mm} \leq s_{21} \leq 920\text{mm}$, $61\text{mm} \leq s_{32} \leq 251\text{mm}$, $-58^\circ \leq \theta \leq 52^\circ$, $-85^\circ \leq \beta \leq 73^\circ$ and $-90^\circ \leq \psi \leq 90^\circ$, respectively. Please note that, for both simulations and experiments, the β angle is taken in the positive range $0^\circ \leq \beta \leq 73^\circ$ to prevent the collision between the arm and the z -axis. The lengths of the manipulator arm links are $l_1 = 90$ mm, $l_2 = 102$ mm and $l_3 = 60$ mm.

An Arduino Mega 2560 board and a Rover5 motor driver are the electrical circuit components of the manipulator platform. For the translational motion, the sensory data is collected from three quadrature encoders (100 cycles/rev and up to 30 kHz data collection frequency) and three motors (gear head DC motor, 12 Volt, a gear ratio of 30:1 and a maximum rotation of 200 rpm) are carried out by the Rover5 motor driver to actuate the platform. For the rotational motion, the Arduino Mega 2560 board provides the execution of three servo motors attached to the revolute joints and one servo motor attached to the end-effector for open-close motion.

5.2 Artificial strawberry plant in experimental set-up

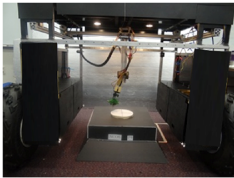
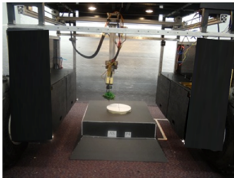

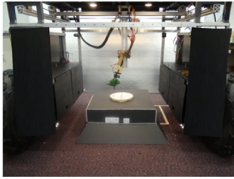
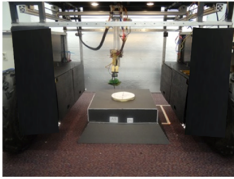
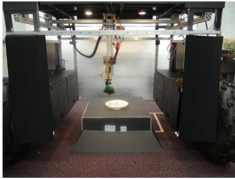
As shown in Figure 7, a white circular plane is used to represent the projected area of a spherical volume covered by a strawberry plant, while the black trapezoidal volume represents a strawberry bed in the field. A plastic plant is put on the circular plane to represent the diseased leaf. The locations of the target leaves are selected such that all four quadrants of the circular plane are tested. Three methods (NCOS, AS, SSAS) are used to calculate the joint commands and these joint commands are then executed according to the command sequence defined in Remark 3.

Figure 7 Experimental set-up and AGR in the laboratory (see online version for colours)

5.3 Experiment results and discussion

A total of 24 experiments (2 target locations from each quadrant per method) were conducted, and for brevity, the final poses of two points $[580, 310, -377.9]$ mm and $[565, 420, -340.1]$ mm are listed in Table 4.

Table 4 Resultant poses of the platform with NCOS, AS and SSAS methods (see online version for colours)

Target Point	NCOS	AS	SSAS
$[580, 310, -377.9]$ mm			
$[565, 420, -340.1]$ mm			

The following observations can be made from the figures in Table 4. All three methods reach the target locations. The final poses calculated by the NCOS are different from those of the AS and SSAS due to the fact that the NCOS compute the optimal solution, while the AS and SSAS provide an analytical solution and an optimal solution within a solution subset, respectively.

Considering the fact that there are sliding, backlash and loosening issues with the belt-pulley system in the mechanism design, the total accuracy of the handling mechanism is within a $[-1, 1]$ cm range.

For the point of $[580, 310, -377.9]$ mm, the execution time of the commands is recorded as roughly 6s in the NCOS, 4s in the AS and 4 s in the SSAS experiments. For

the point of [565, 420, -340.1] mm, the execution time is approximately 4 s in the NCOS, 6 s in the AS and 6 s in the SSAS experiments. The average execution time for the 24 experiments is 5.5 s for the NCOS, 4.95 s for the SSAS and 4.875 s for the AS. The execution times for the points far from the initial reset pose (e.g. the points in the third and fourth quadrants) are higher than the ones closer to the origin.

6 Conclusion

In this study, a set of analytical solutions are developed for a 5-DOF inverse kinematics problem for the handling mechanism of an agricultural robot that will operate in commercial strawberry fields. The minimum energy solution within the solution set is derived. Here the joint constraints are considered in deriving the solution set. The advantages of the proposed algorithms are:

- 1 the computational cost is low
- 2 the optimal solution within the set can be found
- 3 the joint constraints and limitations in the mechanism design are taken into account.

Both simulation and hardware tests verify the effectiveness of the algorithms.

References

- Aristidou, A. and Lasenby, J. (2009) *Inverse Kinematics: A Review of Existing Techniques and Introduction of a New Fast Iterative Solve*. Technical report CUED/F-INFENG/TR-632, University of Cambridge, UK.
- Baerlocher, P. (2001) *Inverse Kinematics Techniques of the Interactive Posture Control of Articulated Figure*, Ph.D. dissertation, Présentée Au Département D'Informatique, EPFL, Lausanne, Switzerland.
- Balestrino, A., De Maria, G. and Sciavicco, L. (1984) 'Robust control of robotic manipulators', in *Proceedings of the 9th IFAC World Congress*, Budapest, Hungary, Vol. 5, pp.2435–2440.
- Bingul, Z., Ertunc, H.M. and Oysu, C. (2005) 'Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist', in Ribeiro, B., Albrecht, R.F., Dobnikar, A., Pearson, D.W. and Steele, N.C. (Eds.): *Adaptive and Natural Computing Algorithms*, Springer, Vienna, pp.112–115.
- Buss, S.R. and Kim, J.S. (2005) 'Selectively damped least squares for inverse kinematics', *Journal of Graphics Tools*, Vol. 10, No. 3, pp.37–49.
- Chirikjian, G.S. (1993) 'A general numerical method for hyper-redundant manipulator inverse kinematics', in *Proceedings of IEEE International Conference on Robotics and Automation*, Atlanta, GA, USA, Vol. 3, pp.107–112.
- Colome, A. and Torras, C. (2015) 'Closed-loop inverse kinematics for redundant robots: comparative assessment and two enhancements', *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 2, pp.944–955.
- Cui, Y., Gejiyama, Y., Kobayashi, T., Hiyoshi, K. and Nagata, M. (2013) 'Study on cartesian-type strawberry-harvesting robot', *Sensor Letters*, Vol. 11, Nos. 6–7, pp.1223–1228.
- David, E.O. and Schrader, W.W. (1984) 'Efficient computation of the jacobian for robot manipulators', *The International Journal of Robotics Research*, Vol. 3, No. 4, pp.66–75.
- Defterli, S.G., Shi, Y., Xu, Y. and Ehsani, R. (2016) 'Review of robotic technology for strawberry production', *Applied Engineering in Agriculture*, Vol. 32, No. 3, pp.301–318.

- Duka, A.V. (2014) 'Neural network based inverse kinematics solution for trajectory tracking of a robotic arm', *Procedia Technology*, Vol. 12, pp.20–27.
- Falco, P. and Natale, C. (2011) 'On the stability of closed-loop inverse kinematics algorithms for redundant robots', *IEEE Transactions on Robotics*, Vol. 27, No. 4, pp.780–784.
- Gan, J.Q., Oyama, E., Rosales, E.M. and Hu, H. (2005) 'A complete analytical solution to the inverse kinematics of the pioneer 2 robotic arm', *Robotica*, Vol. 23, pp.123–129.
- Hajjaj, S.H. and Sahari, K.S.M. (2014) 'Review of research in the area of agriculture mobile robots', in Sakim, M. et al. (Eds.): *8th International Conference on Robotic, Vision, Signal Processing and Power Applications, Lecture Notes in Electrical Eng.*, Malaysia, Springer Science+Business Media Singapore, pp.107–118.
- Hasan, A.T., Al-Assadi, H.M.A.A. and Isa, A.A.M. (2011) 'Neural networks' based inverse kinematics solution for serial robot manipulators passing through singularities', in Prof. Suzuki, K. (Ed.): *Artificial Neural Networks - Industrial and Control Engineering Applications*, InTech, ISBN: 978-953-307-220-3. Available at: <http://www.intechopen.com/books/artificial-neural-networks-industrial-and-control-engineering-applications/neural-networks-based-inverse-kinematics-solution-for-serial-robot-manipulators-passing-through-sing>.
- Jin, Y., Chen, I.M. and Yang, G. (2006) 'Kinematic design of a 6-DOF parallel manipulator with decoupled translation and rotation', *IEEE Transactions on Robotics*, Vol. 22, No. 3, pp.545–551.
- Li, N., Remeikas, C., Xu, Y., Jayasuriya, S. and Ehsani, R. (2015) 'Task assignment and trajectory planning algorithm for a class of cooperative agricultural robots', *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 137, No. 5, p.051004.
- Maciejewski, A.A. and Klein, C.A. (1985) 'Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments', *The International Journal of Robotics Research*, Vol. 4, No. 3, pp.109–117.
- Manocha, D. and Canny, J.F. (1994) 'Efficient inverse kinematics for general 6R manipulators', *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 3, pp.648–657.
- Milella, A., Reina, G. and Foglia, M. (2006) 'Computer vision technology for agricultural robotics', *Sensor Review*, Vol. 26, No. 4, pp.290–300.
- Misaghi, F., Dayyanidardashti, S., Mohammadi, K. and Ehsani, M.R. (2004) 'Application of artificial neural network and geostatistical methods in analyzing strawberry yield data', ASAE paper no. 041147, St. Joseph, MI.
- Mitsi, S., Bouzakis, K.D. and Mansour, G. (1995) 'Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joint limits', *Mechanism and Machine Theory*, Vol. 30, No. 3, pp.653–663.
- Nakamura, Y. and Hanafusa, H. (1986) 'Inverse kinematic solutions with singularity robustness for robot manipulator control', *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, No. 3, pp.163–171.
- Nearchou, A.C. (1998) 'Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified generic algorithm', *Mechanism and Machine Theory*, Vol. 33, No. 3, pp.273–292.
- Novakovic, Z.R. and Nemeč, B. (1990) 'A solution of the inverse kinematics problem using the sliding model', *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 2, pp.247–252.
- Ozgoren, M.K. (2013) 'Optimal inverse kinematic solutions for redundant manipulators by using analytical methods to minimize position and velocity measures', *Journal of Mechanisms and Robotics*, Vol. 5, pp.031009-1–031009-16
- Rodríguez, F., Moreno, J.C., Sánchez, J.A. and Berenguel, M. (2013) 'Grasping in agriculture: state-of-the-art and main characteristics', in Carbone, G. (Ed.): *Grasping in Robotics, Mechanisms and Machine Science*, Springer-Verlag, London, UK, pp.385–409.

- Schmoldt, D. (2012) 'Q&A an US department of Ag expert talks on the national robotics initiative', AUVSI's *Unmanned Systems Mission Critical*, 2(1) [online], pp.26–27. http://issuu.com/auvsi/docs/mission_critical_spring2012S (access 7 November 2016).
- Sciavicco, L. and Siciliano, B. (1988) 'A solution algorithm to the inverse kinematic problem for redundant manipulators', *IEEE Journal of Robotics and Automation*, Vol. 4, No. 4, pp.403–410.
- Shimizu, M., Kakuya, H., Yoon, W., Kitagaki, K. and Kosuge, K. (2008) 'Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution', *IEEE Transactions on Robotics*, Vol. 24, No. 5, pp.1131–1142.
- Slaughter, D.C., Giles, D.K. and Downey, D. (2008) 'Autonomous robotic weed control systems: a review', *Computers and Electronics in Agriculture*, Vol. 61, No. 1, pp.63–78.
- Sumner, R.W., Zwicker, M., Gotsman, C. and Popovic, J. (2005) 'Mesh-based inverse kinematics', *ACM Transactions of Graphics*, Vol. 24, No. 3, pp.488–495.
- Tabandeh, S., Clark, C. and Melek, W. (2006) 'A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering', in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp.1815–1822. DOI:10.1109/CEC.2006.1688527
- Tarokh, M. and Kim, M. (2007) 'Inverse kinematics of 7-DOF robots and limbs by decomposition and approximation', *IEEE Transactions on Robotics*, Vol. 23, No. 3, pp.595–600.
- Tolani, D., Goswami, A. and Badler, N.I. (2000) 'Real-time inverse kinematics techniques for antropomorphic limbs', *Graphical Models*, Vol. 62, pp.353–388.
- Van Henten, E.J., Schenk, E.J., Van Willigenburg, L.G., Meuleman, J. and Barreiro, P. (2010) 'Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot', *Biosystems Engineering*, Vol. 106, No. 2, pp.112–124.
- Waldron, K. and Schmiedeler, J. (2008) '1. Kinematics', *Springer Handbook of Robotics*, Springer, Part A/1, pp.9–33.
- Wang, C., Chung, W., Liao, J., Chung, C., Kuo, Y. and Lin, T. (2012) 'Strawberry anthracnose disease assessment using hyperspectral imaging', in *Proceedings of the 6th International Symposium of Machinery and Mechatronics in Agriculture and Biosystems Engineering*, Jeonju, Korea.
- Wang, F. (2010) 'Guidance line detection for strawberry field in greenhouse', *Paper Presented at the Symposium on Photonics and Optoelectronic*, Chengdu, China, pp.1–4.
- Xu, W., She, Y. and Xu, Y. (2014) 'Analytical and semi-analytical inverse kinematics of SSRMS-type manipulators with single joint locked failure', *Acta Astronautica*, Vol. 105, pp.201–217.
- Xu, Y., Ehsani, R., Kaplan, J., Ahmed, I., Kuzma, W., Olandi, J., Nehila, K., Waller, K. and Defterli, S. (2014), 'An octo-rotor ground network for autonomous strawberry disease detection – year 1 status update', in *2nd International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture and Forestry (RHEA)*, Madrid, Spain, pp.457–466.