Gait planning based on kinematics for a quadruped gecko model with redundancy

Donghoon Son\textsuperscript{a}, Dongsu Jeon\textsuperscript{a}, Woo Chul Nam\textsuperscript{a}, Doyoung Chang\textsuperscript{a}, Tae Won Seo\textsuperscript{b,}\textsuperscript{*}, Jongwon Kim\textsuperscript{a}

\textsuperscript{a} Seoul National University, School of Mechanical and Aerospace Engineering, Seoul, Republic of Korea
\textsuperscript{b} Carnegie Mellon University, Department of Mechanical Engineering, PA, USA

\textbf{ARTICLE INFO}

Article history:
Received 1 July 2009
Received in revised form 12 November 2009
Accepted 24 November 2009
Available online 1 December 2009

Keywords:
Gait planning
Quadruped model
Kinematics
Redundancy
Transition of direction

\textbf{ABSTRACT}

Recent research on mobile robots has focused on locomotion in various environments. In this paper, a gait-generation algorithm for a mobile robot that can travel from the ground to a wall and climb vertical surfaces is proposed. The algorithm was inspired by a gecko lizard. Our gait planning was based on inverse kinematics using the Jacobian of the whole body, where the redundancy was solved by defining an object function for the gecko posture to avoid collisions with the surface. The optimal scalar factor for these two objects was obtained by defining a superior object function to minimize the angular acceleration of joints. The algorithm was verified through simulation of the gecko model traveling on given task paths and avoiding abnormal joint movements and collisions.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Recent research on mobile robots has focused on travel under various conditions such as an irregular terrain \cite{1-4}, inclined/declined planes \cite{5} and stairways \cite{6,7}. Research on overcoming gravity to climb vertical walls \cite{8-11} is another area of interest. Research on robots that can overcome nonstandard ground conditions and climb vertical walls has not been completed.

The gecko lizard can travel under many conditions, including climbing vertical walls. The ability of the gecko to attach to almost any surface using its directional adhesive hair has been studied; however, the gecko has additional advantages such as (1) a sprawled posture that decreases the falling moment, resulting in increased climbing ability, and (2) a flexible but still controllable waist that enables it to move freely from horizontal to vertical planes and vice versa. The gecko is an inspiration to designers of the next generation of mobile robots. We propose a gait-generation algorithm for a robot that is able to travel in many conditions, even on vertical walls.

The related studies have been completed for quadruped mobile robots that travel on the ground. Research on gait generation can be divided into static and dynamic gaits. A dynamic gait is used for rapid locomotion. It balances force and acceleration during movement \cite{12-15}. However, due to control and stability issues, only limited locomotion, such as locomotion on a flat terrain or maneuvering around very small obstacles, is possible. Creating a dynamic gait stepping onto a desired plane with various inclines is a difficult problem to solve. Therefore we focused on a static gait since our goal was the ability to travel on any surface plane and also to freely travel from one surface to another.

A static gait emphasizes gaining stability by placing the center of gravity (CG) inside the supporting polygon, and it is used for walking or other slow movements \cite{16}. Fine control is possible since body motion is directly controlled. Studies on static gaits have occasionally considered how to handle an irregular terrain \cite{5-7} based on studies of locomotion on a flat terrain \cite{17-19}. LittleDog, which is a 12 degrees of freedom (DOFs) point-foot quadruped robot designed and built by Boston Dynamics, Inc., is a remarkable recent product of static gait research. The research focuses on locomotion on an extremely irregular terrain. Research on solving the redundancy of a multi-DOF quadruped robot \cite{20} and improving stability by controlling the trajectory of the CG \cite{2-4} has been completed. The robustness of the control algorithm has been verified by overcoming obstacles as high as its leg height. However, the study cannot be directly applied to static gait planning based on a gecko model. Research on control of the waist and avoidance of collision during movement is necessary for the gecko to travel from the ground to the wall or vice versa.

We developed a gait-generation method for straight and curved path planning on the ground, during wall climbing and for
The redundancy of the multi-DOF system was solved by defining an object function that included the error of the joint angles from empirical data and a real robot design and control by considering both a bio-inspired approach by minimizing errors of considering both abio-inspired approach by minimizing errors and object function into the inverse kinematics equation and inputting the task space command generated by the path/footstep planner.

2.2. Assumptions of kinematic analysis

A quadruped animal needs at least three supporting points to avoid falling over while walking. We also assumed a no-slip condition for a gecko due to its adhesiveness. Following this, our assumptions were:

1. A gecko model always has at least three legs contacting the ground.
2. The feet do not slip.

2.3. Analysis of degrees of freedom

Both closed and open chains were formed at each step of the gecko model. Groubler’s formula was used to define the DOFs of each chain [21].

\[ f = 6(N - 1 - j) + \sum f_i. \]  

(1)

\( f, N, j, f_i \) indicate the total DOFs, number of links, number of joints, and the DOFs of each joint, respectively. The result is given in Table 1.

The total DOFs at each step including both open and closed chains was 16. Normally only three positioning DOFs are given for a task; the remaining three DOFs for rotation are automatically generated. As a result, there were a maximum of 13 redundant DOFs for the joint space for a given task.

2.4. Forward kinematics in the spatial frame

The forward kinematics in the body frame can be derived straightforwardly by evaluating the SE(3) matrix for each joint and multiplying them in order. The body frame was attached to the gecko model and moved as time progressed, making it difficult to describe the movement in space. So we defined the spatial frame as a globally fixed frame.

The traditional method of solving the forward kinematics in a spatial frame is to attach the frame to the ground and assume a leg to be a serial link based on the ground. It is solved by evaluating the angle between the ground and leg and the angles at each joint. However, since the gecko model (quadruped point-foot walker) has no actuator at the foot, the angle between the ground and leg is uncertain, making it impossible to use the traditional method. In this paper we introduce a new method to solve the forward kinematics within a spatial frame to eliminate the need to evaluate the ground contact angle of the leg.

We first defined the fixed legs and swinging leg. Here we chose foot 1 (front left), foot 3 (rear left), and foot 4 (rear right) to be fixed to the ground, and foot 2 (front right) to be the swinging foot. The following procedure holds for every other gait.

The three fixed feet were the only joints that never moved during the gecko model movements. Since the triangle formed by the fixed feet was also always fixed unless the model was altered, the spatial frame was defined based on the triangle (Fig. 3). We could choose any vertex as the origin of the spatial frame and defined one side of the triangle that adjoined it to be the X-axis. The Z-axis was a unit vector normal to the triangle, and the Y-axis was automatically defined. Although any point could be the origin,
we chose the vertex at the front fixed leg (foot 1: front left) to be the origin of the spatial frame.

\[
x^i_2 = \begin{bmatrix} p^i_3 - p^i_1 \\ p^i_3 - p^i_2 \\ \end{bmatrix}, \quad z^i_2 = x^i_2 \times (p^i_2 - p^i_3), \quad y^i_2 = z^i_2 \times x^i_2. \quad (2)
\]

\[
R_{ib} = \begin{bmatrix} x^i_1 & y^i_1 & z^i_1 \\ 0 & 1 & 0 \\ \end{bmatrix}, \quad (3)
\]

\[
T_{ib} = \begin{bmatrix} R_{ib} & -R_{ib} p^i_1 \\ 0 & 1 \\ \end{bmatrix}. \quad (4)
\]

where \( x^i_1, y^i_1, z^i_1 \) denote directional vectors from the spatial frame to the body frame, \( p^i_1 \) denotes a directional vector of the \( i \)th foot in the \( j \)-frame, and \( R_{ib} \) and \( T_{ib} \) denotes an SO(3) and SE(3) matrix from the \( i \)th frame to \( j \)th frame, respectively.

The triangle was altered when the step was changed during walking. Since the spatial frame was defined based on the triangle at a step, in order to have a global frame we needed a method to interpret the frames at each step with respect to the initial spatial frame.

At the instant the swinging foot touched the ground, all four feet were on the ground. The frame of the next step with respect to the current frame could be defined at this moment by evaluating the SE(3) matrix between the two frames.

\[
T_{i(i+1)} = T_{i(ib)}(T_{i(i+1)bn})^{-1}. \quad (5)
\]

By serially multiplying the matrices of (4) we can define any frame with respect to the initial spatial frame.

\[
T_{ibn} = T_{ib1} T_{1b2} T_{2b3} \cdots T_{(n-1)b(n-1)} \times (T_{nbn})^{-1} T_{nbn}. \quad (6)
\]

By the definition of spatial frame, the forward kinematics of foot 2 (swing foot) with respect to the spatial frame could be derived.

\[
p^0_2 = T_{ibn} p^p_{ib}. \quad (7)
\]

2.5. Jacobian analysis

2.5.1. Constraint Jacobian

The closed parallel chain of the model has three dependent joints. The values of these joints were obtained from the constraint that three legs are fixed to the ground. These three fixed legs give us nine constraint equations in a spatial frame, but the effect of 6-DOF spatial motion of the base frame should be ignored in relative kinematics of the model. And so, only three constraints of the constant distance between each foot define the constraint equation, as follows:

\[
|p_i(q_{up}, q_u) - p_j(q_{up}, q_u)|^2 = g_{i}(q_{up}, q_u) = \text{const}, \quad (8)
\]

where \( i, j = 1, 3, 4 \) and \( i \neq j \), \( p_i \) is the position of the \( i \)th foot, and \( q_{up} \) and \( q_u \) are the independent and dependent joints of the closed parallel chain, respectively.

Deriving the constraint equation using (8) is complicated, so we used the partial derivative with respect to time. Differentiating \( g_i \) with respect to time, we obtain

\[
\frac{\partial g_i}{\partial q_{up}} + \frac{\partial g_i}{\partial q_u} = 0, \quad (9)
\]

where \( i = 1, 3, 4 \). By defining \( G_{up,i} = \frac{\partial g_i}{\partial q_{up}} \), the equation becomes

\[
G_{up,i} p_i + G_u q_u = 0, \quad (10)
\]

where \( i = 1, 3, 4 \). Substituting \( G_{up} = [G_{up,1} G_{up,3} G_{up,4}]^T \) and \( G_u = [G_1, G_3, G_4]^T \), the equation becomes

\[
[G_{up} G_u] \begin{bmatrix} q_{up} \\ q_u \end{bmatrix} = 0. \quad (11)
\]

Since the inverse of \( G_u \in \mathbb{R}^{12 \times 12} \) exists, we expand the equation and multiply each side by \( G_u^{-1} \):

\[
\dot{q}_u = G_u^{-1} G_{up} \ddot{q}_{up} = \Phi (q_{up}) \ddot{q}_{in} = \Phi \dot{q}_{in}. \quad (12)
\]

where \( q_{in} \in \mathbb{R}^4 \) is the joint value of the open serial chain and \( q_u \in \mathbb{R}^{16} \) is the independent joint value. To derive the constraint Jacobian, \( \Phi \in \mathbb{R}^{3 \times 16} \), between \( q_u \) and \( q_{in} \) in (12), we insert a zero column into the column of \(-G_u^{-1} G_{up} \in \mathbb{R}^{3 \times 12}\) that corresponds to \( q_{in} \). The constraint Jacobian is used to derive the forward Jacobian in the next section.

2.5.2. Forward Jacobian of whole body

The forward Jacobian of the 16 independent joints was obtained by differentiating (7) with respect to the independent joints \( q_u \) by definition:

\[
J = J(q_u) = \frac{\partial p_2^i(q_u, q_u)}{\partial q_u} = \frac{\partial p_2^i(q_u, \Phi q_u)}{\partial q_u} \in \mathbb{R}^{3 \times 16}. \quad (13)
\]

where \( p_2^i \) is the swing foot position on the spatial frame from (7), \( q_u \) is the independent joints, \( \Phi \) is the constraint Jacobian and \( J(q_u) \) is the forward Jacobian.

3. Gait planning based on inverse kinematics

The procedure of gait planning is shown in Fig. 4. First, footsteps are generated in various moving conditions such as transitions and steering. Then foot swing trajectory connecting each footstep is generated smoothly based on a spline curve. Finally, the joint angles of gecko model satisfying the constraints of the feet positions are determined by solving an optimal problem using redundant parameters based on kinematics. This section explains the optimal joint angle determination problem and swing trajectory generation.
Fig. 4. Procedure of kinematic-based gait planning.

3.1. Framework

The solution of the inverse kinematics for a redundant system is not unique. To find the optimum solution we used the null-space optimization method as in [22]:

\[
q = J^T \dot{X} + \gamma (I - J^T J) \cdot \nabla_q H(q),
\]

where \( J \) is the forward Jacobian of whole body, from (12), \( J^+ = (J^T J)^{-1} J^T \) is the Moore–Penrose pseudoinverse of \( J \), \( \gamma \) is a scalar factor and \( H(q) \) is the object function. By projecting the gradient of the object function to the null-space of the forward Jacobian, we obtained a solution that minimized \( H(q) \) while completing the task.

3.2. Definition of the object function

We chose the following two conditions as criteria for an optimum motion.

1. A motion similar to real gecko movement: minimizing the error of the joint angles from a reference posture.
2. A motion that does not collide with the ground or wall: collision avoidance.

The first condition works to make the generated gait resemble a real gecko’s gait, and the second condition works so as to not collide with the ground during locomotion. These two conditions were defined as the object function

\[
\min H(q) = \alpha \left\{ \sum_{i=1}^{19} (q_i - q_{i,ref})^2 \right\} + \beta \left\{ \sum_{i=1}^{3} (h_i - h_{i,ref})^2 \right\},
\]

where \( q_{i,ref} \) and \( h_{i,ref} \) denote the reference joint angles and reference heights of three joints of the body, respectively.

The first term of the object function represents minimizing the error of the joints. It focuses on minimizing the joint angle differences from the experimental values of a real gecko. Since the object of the function was to imitate gecko movement (Fig. 5(a)), impossible joint angles were avoided.

From the measured data of the gecko’s locomotion explained in Section 3.3, reference postures should be determined. Moving the reference posture as a function of time is not a good idea since the exact synchronous implementation is a difficult problem in various moving conditions. Instead of moving the reference posture, we used two static reference postures, presented in Fig. 5(a), to generate the locomotion. Switching of reference postures during locomotion is required, and the condition of switching is determined by the contact of feet with the ground. And so it can be easily implemented on the entire locomotion of the model.

The second term is the collision avoidance term. By keeping the distance of the body (neck, waist, bottom) to the ground constant, we avoided collision with the surface (Fig. 5(b)). The values \( h_1, h_2, h_3 \) are also determined from empirical data of the locomotion of a gecko.

3.3. Measurement of reference joint angles

Two reference values in (15), \( q_{i,ref} \) and \( h_{i,ref} \), were determined by empirical data measured from a real gecko’s locomotion. Details of the procedure of measuring the reference values were presented in previous research of the authors [23]. The posture of the gecko was measured using two camcorders for side views and one camcorder for an upper view. The positions of the joints were marked with white color and the measured position data of the joints were extracted from the image files manually using a CAD program. By using the position data of each joint, each angle of the joints could be calculated using geometrical constraints.

3.4. Selection of the optimum scalar factor for the object function

The scalar weighting factor \((\alpha, \beta)\) needed to be chosen since two object functions (15) were linearly combined. Since the locomotion of the gecko model is dependent on \((\alpha, \beta)\), we used a method to obtain the optimum values.

First we defined a function that was the total sum of the angular acceleration of the gecko model for a given path. The optimized function produced values to be minimized for optimal performance, such as the required motor torque, system energy, and vibration produced, which are critical constraints in a real machine design. Since the values are proportional to the function,
we set it as a superior object function to be minimized and found the values of \((\alpha, \beta)\) that satisfied, as follows:

\[
\min f(\alpha, \beta) = \int_0^t \left( \sum_{i=1}^{19} |\dddot{q}_i(\alpha, \beta)| \right) dt
\]

along a predefined path. (16)

We should define a path to determine \((\alpha, \beta)\) based on the optimal problem in (16). The path should contain every considerable path along which the gecko model might move. The predefined path to be used to determine the weighting factors is shown in Fig. 9; it consists of flat/vertical walking, internal/external transitions, inclined/declined walking, and steering. The determined weighting factors along this predefined path are expected as robust values that can work in various moving conditions.

The optimum values of \((\alpha, \beta)\) were found using the gradient descent method and golden section search method [24,25].

3.5. Generating the swing trajectory of the foot

In each step, the gecko model first chooses its next footstep position and then swings its leg to that position. In order to accomplish this movement, a swing trajectory needed to be generated.

When a gecko swings its foot outwards, its body follows. We first created a new plane by rotating 90° outward the plane that the current footstep point \((p_s(t_0))\), next footstep point \((p_s(t_0 + t_{interval}))\) and neck point \((p_{neck})\) was forming (Fig. 6). An isosceles triangle was made on this plane whose bottom side was \(p_s(t_0 + t_{interval}) - p_s(t_0)\) and height was \(d_{swing}\). The vertex of the triangle was chosen to be the intermediate footstep position \((p_s(t_0 + 0.5 \times t_{interval}))\).

The swing trajectory of the foot was the spline created by the three points: the current footstep, the intermediate footstep and the next footstep.

3.6. Joint space control

In order to substitute trajectory planning into the framework, we evaluated the velocity at each trajectory point using

\[
dp_s = p_s(t + dt) - p_s(t),
\]

where \(p_s(t)\) is the position of the foot. Substituting the object function and task into the framework, we obtained the joint space command for a discrete time \(dt\):

\[
dq = J^+ dip_s + \gamma (I - J^+J) \cdot V_q H(q),
\]

where \(J\) is the forward Jacobian of the whole body, \(J^+\) is the Moore–Penrose pseudoinverse of \(J\) from (13), is the task space
command from (17), $\gamma$ is a scalar factor and $H(q)$ is the object function from (15).

By controlling the joint space with the solution from (18) the gecko model obtained a minimum angular acceleration control input that both satisfied the given task and minimized the object function.

4. Simulation under various movement scenarios

4.1. Generation of footsteps

Some rules are set for the footprint generation of the walking gecko model. First, one stride should be the distance the gecko model travels during one cycle, which is shown in Fig. 7. Next, the footprint position set for a straight path should be symmetric every half cycle. The sequence of the gait is determined based on the measurements [23]. In order to realize this, the following conditions should be satisfied:

1. The shape of the initial footprint position set is a parallelogram.
2. One foot moves the length of one stride every $1/4$ of a cycle.

During internal transitions, the fore feet should move from the horizontal to the inclined surface while the head maintains its position to prevent collision. During external transitions, the head should move forward to enable the transition while the feet hold their positions. And so, the footprint position set for an inclined/declined path required a footprint correction of $2h \tan \left(\frac{\theta}{2}\right)$ due to the distance difference of the body and foot trajectory. The value, $2h \tan \left(\frac{\theta}{2}\right)$, is determined by the geometrical relation shown in Fig. 7(b).

The straight path footprint, curved path footprint, inclined footprint and declined footprint can be generated based on this footprint rule. The straight path footprint was generated by shifting the initial footprint one stride for each cycle (Fig. 8). The curved path footprint was generated by bending the initial footprint position of the straight path with a desired radius $R$ and then rotating it one stride for each cycle with respect to the center of curvature. The inclined path footprint was generated by inserting $2h \tan \left(\frac{\theta}{2}\right)$ of blank space into the straight path footprint where the incline started. Similarly, the declined path footprint was generated by inserting footsteps for length $2h \tan \left(\frac{\theta}{2}\right)$ into the straight path footprint where the decline started.

4.2. Stability

The static stability of legged locomotion on a horizontal plane is determined by the relation between the positions of the supporting
Internal transition to vertical plane
External transition to vertical plane
Horizontal plane to 30° declined plane
30° declined plane to horizontal plane
Curved path (R 150mm)

Fig. 9. Simulation result under various moving scenarios.
Feet and the position of the center of gravity (COG). To maintain static stability when walking on a normal plane the COG should be located inside the closure of the three supporting legs. This is avoid the moment of the body making the robot loose stance and crash into the ground. However, in the case of a gecko robot that has an attachment force in the feet, the problem is slightly different. That is, the attachment force in the feet produces additional forces to overcome the pitching moment of the body.

The static stability during climbing vertical walls or ascending or descending inclined planes is guaranteed by two conditions. First the total attachment force should be larger than its body mass. Second the attachment force in the fore foot should be able to resist the pitch-back moment of the body which is critical in the gait when the robot is changing one of its fore feet’s position, leaving only one fore foot attached.

In both cases static stability could be obtained by producing enough attachment force in the feet with respect to the body mass, especially for the fore feet. Since the gait simulation presented in this paper is based on the simulation, the stability cannot be determined without assuming masses and attaching forces. The stability analysis result would give us the required attaching forces of the feet for a specific robot to walk and climb stably.

4.3. Simulation results

Footstep generation was simulated using the gait-planning algorithm. The test path included all conditions the gecko model could encounter, such as straight/curved and inclined/declined paths. The scalar factors were optimized for the given path. Consequently, the gecko model movement was similar to that of a real gecko, avoiding abnormal joint movements and collisions with the ground. Smooth movement was generated with no vibration or jerking (Fig. 9).

The locomotion from the horizontal to the vertical plane was checked to verify the performance of the collision avoidance term in the object function. The results with and without the avoidance term were checked (Fig. 10). Both simulations consisted of the same footstep and swing foot trajectories; however, the results varied depending on whether the neck, waist, and hip were controlled or not. Collision with the surface occurred when the collision avoidance term was not included. However, collision was avoided with the term added.

The effect of the scalar factor optimization was determined by comparing the simulation results before and after optimization. The angular acceleration was compared for a given path (first slide of Fig. 9) and the resulting accelerations are shown in Fig. 11. The results without optimization (initial scalar factor used = 0.5/0.5) show more fluctuation, which results in more vibration and larger starting torque requirements. Also, since the maximum acceleration is larger, the maximum required torque increases, which means that a bigger actuator is needed. In contrast, after optimization, the motion was much smoother, which means better durability and stability. The maximum joint angular acceleration was decreased by 36.43% and the sum of the joint angular acceleration was decreased by 64.65%. A smaller actuator is better for a mobile climbing robot.

5. Conclusion

This paper contains gait-planning calculations for a robot modeled on a gecko. A spatial frame was defined that was global to any frame generated by three feet, without considering the contact angle. The framework was divided into serial and parallel parts and was unified after separate analyses. An object function was defined to solve the uncertainty of the inverse kinematics of the redundant system. A minimizing error term based on real gecko movement and a collision avoidance term were defined in the object function for smooth and natural movements. A superior object function was defined to optimize the scalar factors in the object function. The gate generation based on swing foot trajectory and footstep generation rules was simulated and verified.

There are several remaining problems as future work. To be used on non-flat terrains, the proposed algorithm should be combined with a reflex control algorithm which can adapt to various irregular terrains. Since a different reference posture is required for another robot kinematic model, finding the proper reference posture might be another challenging problem to be researched. And dynamic analysis to guarantee stability is essential, especially for climbing robot applications.
Acknowledgements

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government [NRF-2009-352-D00018] and Brain Korea 21 Project.

References


[19] Dongsoo Jeon received his B.S. degree in mechanical engineering and mathematics from Seoul National University, Seoul, Korea, in 2004, where he is currently pursuing a Ph.D. degree. His research interests include the design and motion planning of robots. He is currently completing his military service in the armed forces of the Republic of Korea.

[20] Dongboon Son received his B.S. and M.S. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, Korea, in 2007 and 2009, respectively. His research interests include the design and motion planning of robots. He is currently completing his military service in the armed forces of the Republic of Korea.

[21] Woo Chul Nam received his B.S. degree in 2007 and M.S. degree in 2009 at the Robust Design Engineering Laboratory, all from Seoul National University and in mechanical engineering. His current research interests include locomotion of the gecko, pairs of crawling mechanisms, legged robotics and climbing robots.

[22] Doyoung Chang received his B.S. degree in mechanical engineering and mathematics from Seoul National University, Seoul, Korea, in 2004, where he is currently pursuing a Ph.D. degree. His research interests include bio-inspired robot design and industrial welding carriage robots.

[23] TaeWon Seo is post-doctoral researcher of Nanorobotics laboratory in Department of Mechanical Engineering, Carnegie Mellon University. He received the B.S. and Ph.D. degree in the School of Mechanical and Aerospace Engineering from Seoul National University (SNU), Seoul, Korea, in 2003 and 2008. He was with BK21 School of Creative Design of Next Generation Mechanical and Aerospace Systems in SNU as a post-doctoral researcher from 2008 to 2009, and in the same period, he was also with Institute of Advanced Machinery and Design in SNU as a post-doctoral researcher. His current research interests include mechanism design, optimal design, control, and motion planning of robotic systems.

[24] Jongwon Kim is a professor in the School of Mechanical and Aerospace Engineering of Seoul National University, Korea. He received his B.S. in Mechanical Engineering from Seoul National University in 1978, and his M.S. in Mechanical and Aerospace Engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1980. He received his Ph.D. in Mechanical Engineering from the University of Wisconsin–Madison, USA in 1987. He worked with Daewoo Heavy Industry & Machinery, Korea, from 1980 to 1984. From 1987 to 1989, he was Director of the Central R&D Division at Daewoo Heavy Industry & Machinery. From 1989 to 1993, he was a researcher at the Automation and Systems Research Institute at Seoul National University. His research interests include parallel mechanisms, Taguchi methodology and field robots.