

# A whole-body rescue motion control with task-priority strategy for a rescue robot

Wonsuk Lee<sup>1</sup> · Youngwoo Lee<sup>1</sup> · Gyuhyun Park<sup>1</sup> · Seongil Hong<sup>1</sup> · Younsik Kang<sup>1</sup>

Received: 8 July 2015 / Accepted: 17 March 2016 / Published online: 31 March 2016  
© Springer Science+Business Media New York 2016

**Abstract** This paper introduces a new rescue robot consisting of dual-manipulator and variable configuration mobile platform for multi-purpose such as casualty extraction and hazardous goods transport. A specific rescue motion strategy using a whole-body is suggested to tackle characteristics of the robot configuration and balancing issue. In order to take into account safety and stability of the robot during the rescue motions, some restrictions are reflected into redundant domain of the robot with different priority. For stable motion control in various scenarios, a singularity-robust inverse kinematics is adopted and modified to induce smoother robot movement. The robustness of the control approach is checked numerically by comparing other method and experiments for the rescue motion strategy are carried out by using a small-scaled simulator in place of the rescue robot under development.

**Keywords** Rescue robot · Whole-body motion control · Singularity robust inverse kinematics · Closed-loop inverse kinematics (CLIK)

## 1 Introduction

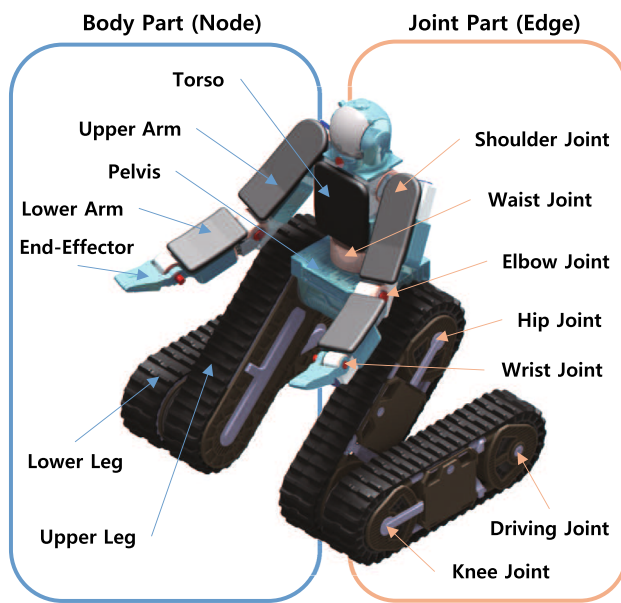
Since accidents and disasters that are beyond human control have been occurred frequently, rescue operation by a robot in place of humans has been spotlighted recently. Thanks to the needs and the technological advancement, therefore, a rescue robot is under active research and development all over the world. In accordance with the research trend, we have been

developing a new rescue robot system shaped like a half-humanoid for multiple purposes such as casualty extraction and explosive transport in hazardous environment. The upper body of the robot consists of a waist and dual-manipulator for better dexterous management and the lower body is a variable configuration platform made of two legs covered with caterpillar track for better cross-country mobility as shown in the Fig. 1. Movement of the robot is performed primarily by driving rather than locomotion as shown description in the Fig. 2.

Since the robot can utilize the whole-body for any end-effector (EE) operation, the redundancy problem has to be tackled for the inverse kinematics (IK). There are many approaches to solve the such IK of the manipulator with redundant degree of freedoms (DOFs). A typical approach is a closed-loop inverse kinematics (CLIK) based on differential kinematics using pseudo-inverse of a Jacobian matrix (Siciliano 1990a; Siciliano et al. 2009). In order to avoid obstacle/constraints or prevent a self-motion due to redundancy, subtasks are imposed to null space of a primary task with lower priorities (Nakamura et al. 1987; Sciavicco and Siciliano 1988; Siciliano 1990b; Siciliano and Slotine 1991). This approach has a drawback related with joint divergence due to the well known kinematic/algorithmic singularities; they are referred in the case of losing a DOF in terms of single or multiple-task, respectively (Siciliano and Slotine 1991). On the other hand, the IK through optimization process based on not the linear differential kinematics but the nonlinear forward kinematics has been briskly researched since it has no singularity issues thanks to not using a Jacobian (Wang and Chen 1991; Mitsi et al. 1995; Kanoun et al. 2011). Besides a local minimum, however, its application could be limited in real-time operation due to computational burden. For real-time implementation, this study adopts an approach like the typical CLIK with task-priority strategy (TP-CLIK) and

✉ Wonsuk Lee  
wsblues82@gmail.com

<sup>1</sup> Agency for Defense Development, Daejeon,  
Republic of Korea



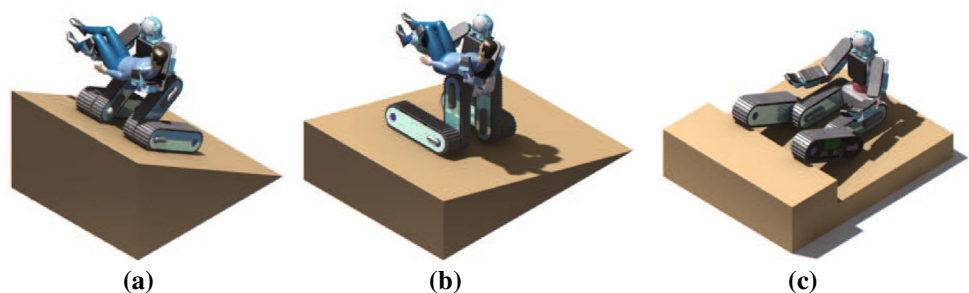
**Fig. 1** Concept of a rescue robot

discusses the existing two remedies for the algorithmic singularity as well as the kinematic singularity.

To deal with the kinematic singularity, there are general two approaches; generating avoidance command by a null motion through a potential field (Siciliano et al. 2009; Yoshikawa 1984) and perturbing an inversion matrix, referred as a damped least-square (DLS) method (Nakamura 1986; Maciejewski and Klein 1988; Chiaverini et al. 1991). Because we will restrict all DOFs subjecting subtasks, however, the latter is more suitable to our approach.

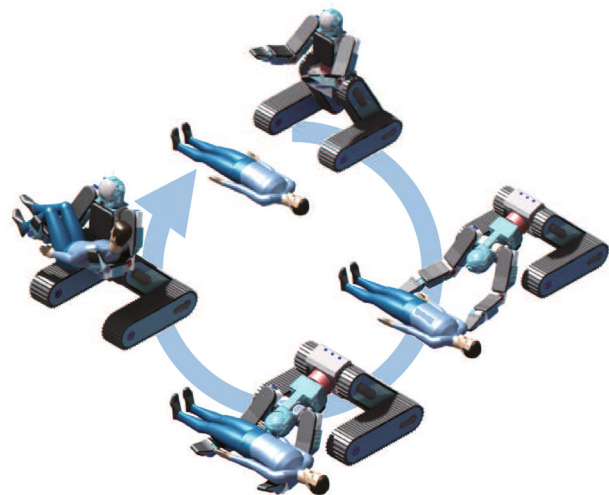
The DLS method can be applied to handle the algorithmic singularity too. Since the rescue robot will be operated with different missions in various environment the IK algorithm considering multiple-task has to robustly generate joint solutions under any circumstance. However, it is not easy to select proper a perturbation parameter of the DLS in order to satisfy both task accuracy and robustness in accordance with various mission scenarios, which will be analyzed numerically in Sect. 4. Therefore, this paper adopts the singularity-robust inverse kinematics approach (Chiaverini 1997) and suggests a robust TP-CLIK (RTP-CLIK) based on modification of the

**Fig. 2** Concept of a rescue robot movement by driving. **a** Longitudinal slope, **b** latitudinal slope, **c** obstacle



approach with 2nd-order expression for smoother joint movement. We check the applicability of the algorithm through experiments as well as numerical simulations. Also, an inherent undesirable end-motion induced by the RTP-CLIK was found and a new motion smoother is introduced to handle the problem.

This paper focuses on the only rescue motions among the various missions. A specific behaviors as described in the Fig. 3 are proposed because feasible rescue motions to approach and lift a casualty are restricted due to the robot configuration as well as balancing problem in the case of lifting situation. It can be easily anticipated that other postures cause rollover of the robot. The motions consist of five primitives which are lowering, forwarding, contact, backwarding, lifting phases sequentially. To prevent rollover of the robot and collision between the robot and the ground during the motions, corresponding subtasks have to be reflected simultaneously and it is realized by using the suggested algorithm subjecting them into the null space of primary task. Since each task could be defined based on different reference coordinates, a Jacobian matrix of each task expressed in the body frame has to be transformed in accordance with assigned coordinates, which will be dealt with in Sect. 3. While many researches on the balancing control of a humanoid applied a



**Fig. 3** Concept of the proposed consecutive rescue motions

zero moment point (ZMP) to keep stability even for dynamic motion (Vukobratovic and Borovac 2004; Kajita et al. 2003; Sardain and Bessonnet 2004; Kim and Oh 2013), a center of gravity (COG) is adopted in this paper since the rescue motion is carried out slowly enough to ignore the dynamic stability. Furthermore, a support polygon of the robot is wide enough thanks to the lower body configuration. The above rescue motion strategy is proved by real experiments utilizing a small-scaled simulator because the rescue robot is still under development.

## 2 Inverse kinematics for redundant robotic systems

The forward kinematic mapping for robot manipulators can be written as

$$\mathbf{x} = f(\mathbf{q}) \tag{1}$$

where  $\mathbf{x} \in \mathbb{R}^r$ ,  $\mathbf{q} \in \mathbb{R}^n$  denote states of task and joint space respectively. Since  $f$  of the above equation is highly nonlinear function, however, reconfigured forward kinematics in velocity domain, which has linear relationship between task and joint space, is usually applied for the IK problem.

$$\dot{\mathbf{x}} = \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{2}$$

Though there are various approaches for the IK, the CLIK algorithm is generally utilized and the 2nd-order equation for smooth motion is expressed as follows: (Siciliano et al. 2009)

$$\ddot{\mathbf{q}} = \mathbf{J}^\dagger(\ddot{\mathbf{x}}_d + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I}_n - \mathbf{J}^\dagger\mathbf{J})\ddot{\mathbf{q}}_0 \tag{3}$$

where  $\ddot{\mathbf{x}}_d$ ,  $\mathbf{e}$  represent desired acceleration and error in operational space, respectively.  $\mathbf{K}_D$ ,  $\mathbf{K}_P$  are  $n \times n$  positive definite matrices and  $\mathbf{I}_n$  means  $n \times n$  identity matrix. A superscript ‘†’ represents right pseudo-inverse for a non-square matrix. Note that the last term in right-hand side of the Eq. (3) can be adopted only in the case of redundant manipulators, i.e.  $r < n$ .  $\ddot{\mathbf{q}}_0$  can be arbitrary joint accelerations and a gradient projection method (GPM) is generally used.  $(\mathbf{I}_n - \mathbf{J}^\dagger\mathbf{J})$  means a projector onto null space of the Jacobian.

### 2.1 TP-CLIK in a recursive form

In the case of highly redundant systems, multiple subtasks with different task-priority can be subjected based on the property of null space of Jacobian for the IK. Let  $\dot{\mathbf{x}}_i = \mathbf{J}_i\dot{\mathbf{q}}$  denote  $i$ -th task kinematics, then a 2nd-order recursive formula of the TP-CLIK is expressed as follows: (Siciliano and Slotine 1991)

$$\begin{aligned} \ddot{\mathbf{q}}_i &= \ddot{\mathbf{q}}_{i-1} + \tilde{\mathbf{J}}_i^\dagger(\ddot{\mathbf{x}}_i + \mathbf{K}_{Di}\dot{\mathbf{e}}_i + \mathbf{K}_{Pi}\mathbf{e}_i - \dot{\tilde{\mathbf{J}}}_i\dot{\mathbf{q}}_{i-1} - \tilde{\mathbf{J}}_i\ddot{\mathbf{q}}_{i-1}) \\ &\quad + \tilde{\mathbf{J}}_i^\dagger(\dot{\mathbf{x}}_i - \mathbf{J}_i\dot{\mathbf{q}}_{i-1}) \\ \ddot{\mathbf{q}}_1 &= \mathbf{J}_1^\dagger\ddot{\mathbf{x}}_1 + \mathbf{J}_1^\dagger\dot{\mathbf{x}}_1 \end{aligned} \tag{4}$$

where

$$\tilde{\mathbf{J}}_i = \mathbf{J}_i\mathbf{N}_{A,i-1} \tag{5}$$

$$\mathbf{N}_{A,i-1} = (\mathbf{I}_n - \mathbf{J}_{A,i-1}^\dagger\mathbf{J}_{A,i-1}) \tag{6}$$

Here,  $\mathbf{J}_{A,i}$  denotes augmented Jacobian

$$\mathbf{J}_{A,i} = [\mathbf{J}_1^T \mathbf{J}_2^T \dots \mathbf{J}_i^T]^T \tag{7}$$

The time-derivative terms,  $\dot{\tilde{\mathbf{J}}}_i^\dagger$ ,  $\dot{\tilde{\mathbf{J}}}_i$ , can be written as following explicit formulas.

$$\dot{\tilde{\mathbf{J}}}_i^\dagger = -\tilde{\mathbf{J}}_i^\dagger\dot{\tilde{\mathbf{J}}}_i\tilde{\mathbf{J}}_i^\dagger + (\mathbf{I}_n - \tilde{\mathbf{J}}_i^\dagger\tilde{\mathbf{J}}_i)\dot{\tilde{\mathbf{J}}}_i^\dagger(\tilde{\mathbf{J}}_i\tilde{\mathbf{J}}_i^T)^{-1} \tag{8}$$

$$\dot{\tilde{\mathbf{J}}}_i = -\mathbf{J}_i(\mathbf{Q}_{A,i-1} + \mathbf{Q}_{A,i-1}^T) + \mathbf{J}_i\mathbf{N}_{A,i-1} \tag{9}$$

where

$$\mathbf{Q}_{A,i} = \mathbf{J}_{A,i}^\dagger\dot{\mathbf{J}}_{A,i}\mathbf{N}_{A,i} \tag{10}$$

Note that a joint solution of the Eq. (4) could becomes divergent, which is so-called an ‘algorithmic singularity’, when  $\tilde{\mathbf{J}}_i$  loses rank. It could be occurred whenever  $\mathbf{N}(\mathbf{J}_{A,i-1}) \cap \mathbf{N}(\mathbf{J}_i) \neq 0$ , meaning that both task spaces are not independent (Chiaverini 1997). In this case, the numerical solution gets divergent or ill-conditioned  $\tilde{\mathbf{J}}_i$  can cause excessive joint acceleration, so we have to treat the problem carefully whenever many subtasks are subjected. A DLS method which introduces a perturbation into  $\tilde{\mathbf{J}}_i^\dagger$  is a general approach for this issue like an approach for the kinematic singularity. Since a perturbation can distort a primary task as well as subtasks, however, it has to be selected cautiously to satisfy both accuracy and robustness. It is not easy to set a proper value for a perturbation taking into account various situations and this limit gets worse as more subtasks are reflected simultaneously.

### 2.2 Singularity robust TP-CLIK

To handle the algorithmic singularity issue more robustly, Chiaverini suggested the new RTP-CLIK as follows: (Chiaverini 1997)

$$\dot{\mathbf{q}} = \mathbf{J}_h^\dagger\dot{\mathbf{x}}_h + \mathbf{N}_h\mathbf{J}_l^\dagger\dot{\mathbf{x}}_l \tag{11}$$

As shown in the Eq. (11), there is no more concern about the algorithmic singularity since  $\tilde{\mathbf{J}}_l^\dagger$  is eliminated by sepa-

rating  $\mathbf{N}_h$  and  $\mathbf{J}_l^\dagger$ . This approach seems to have an intuitive structure to solve the IK because the inversion for each task is decoupled individually and the joint solution for the lower task is projected directly onto the null space of the hightask Jacobian. (Refer to the Appendix for detailed analysis on the algorithmic singularity)

However, there are some drawbacks of the suggested robust algorithm. It cannot generate an exact optimized solution because the inversion is applied in wrong subspace which does not take into account the effect by higher task (Baerlocher and Boulic 1998). Also, it cannot guarantee error convergence of subtasks to zero while a primary task is always satisfied. For error convergence problem, an online-gain tuning method was introduced (Antonelli 2009), but some risk could be accompanied in the case of real application. Whenever many tasks with different priority are considered simultaneously, therefore, we have an opportunity to choose one of both approaches in accordance with given situations. In the case of our studies, since a rescue robot has to be operated in various environments with diverse missions, it is not easy to predict how a primary task conflicts with lower tasks and to select a proper perturbation value for the DLS approach. For these reasons, therefore, we decided that the robust algorithm is more proper to focus on guarantee of the stability of the IK against the algorithmic singularity and tracking accuracy of a primary task even if it suffers lack of satisfaction of lower tasks.

In order to move joints more smoothly, this paper introduces a new 2nd-order equation of the robust algorithm Eq. (11) with the CLIK formulation in generalized form as follows:

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_1 + \sum_{i=2}^m (\dot{\mathbf{N}}_{A,i-1} \dot{\mathbf{q}}_i + \mathbf{N}_{A,i-1} \ddot{\mathbf{q}}_i)$$

$$\ddot{\mathbf{q}}_i = \mathbf{J}_i^\# (\ddot{\mathbf{x}}_{di} + \mathbf{K}_{Di} \dot{\mathbf{e}}_i + \mathbf{K}_{Pi} \mathbf{e}_i - \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i) \quad (12)$$

where  $m$  denotes the number of subtasks,  $\mathbf{q}_i$  means joint space of each individual  $i$ -th task domain and  $\dot{\mathbf{N}}_A$  can be calculated numerically. Because the IK based on the 2nd-order equation generates solutions with velocity-continuity, the resultant motion can be smoother and more natural. Especially, such characteristics of the robot motion is important for the rescue mission to safely handle a casualty. The CLIK approach is adopted to compensate the subtask error derived by the projection onto higher null space. Besides, in order to tackle the kinematic singularity as well as the algorithmic singularity, the DLS inverse denoted as ‘#’ is adopted instead of the pseudo-inverse for inversion problem of corresponding Jacobian of each task.

$$\mathbf{J}_i^\# = \mathbf{J}_i^T (\mathbf{J}_i \mathbf{J}_i^T + \lambda^2 \mathbf{I}_n)^{-1} \quad (13)$$

where  $\lambda \in \mathbb{R}$  represents a perturbation. Since it is possible to distort task performance, specially even in the outside of the vicinity of the kinematic singularity, however, the  $\lambda$  is considered to be varied according to a minimum singular value as a criterion as follows: (Chiaverini et al. 1991)

$$\lambda^2 = \begin{cases} \left(1 - \left(\frac{\sigma_m}{\epsilon}\right)^2\right) \lambda_M^2 & \text{if } \sigma_m < \epsilon \\ 0 & \text{if } \sigma_m \geq \epsilon \end{cases} \quad (14)$$

where  $\lambda_M$  denotes an amplitude of the maximum perturbation at the singularity and  $\epsilon$  is a threshold set based on numerical simulations. Thanks to this approach, the IK solution can be continuous at the vicinity of a kinematic singularity with the varying perturbation from 0 to  $\lambda_M$ .

Here, note that the suggested approach could not converge the error of subtasks to the zero as mentioned previously. Though the CLIK formulation can help reduce the error of subtasks, this could cause an unnatural end-motion of the robot, which seems to be an inherent problem of the RTP-CLIK. It is because the null motion according to the subtasks are not vanished at the final moment due to non-zero tracking errors of the subtasks. To eliminate the phenomenon, we introduce a simple motion smoother which leads the joint velocities to the zero during quite short interval ( $\Delta$ ) at the vicinity of the end of motion. This approach could perturb the tracking error of the primary task, but it would not be significant because the original error order of the primary task as well as the time interval is small enough. For the motion smoother, a ‘1 + cos’ function is utilized as following:

$$\dot{\mathbf{q}}(t) = \begin{cases} \frac{\dot{\mathbf{q}}(t)}{2} \left(1 + \cos\left(\frac{\pi(t-t_0)}{t_f-t_0}\right)\right) & \text{if } t \geq t_0 \\ \dot{\mathbf{q}}(t) & \text{if } t < t_0 \end{cases} \quad (15)$$

where  $t, t_f$  mean the current and final time of task,  $t_0$  is the start time of the smoother setting by  $t_f - \Delta$ .

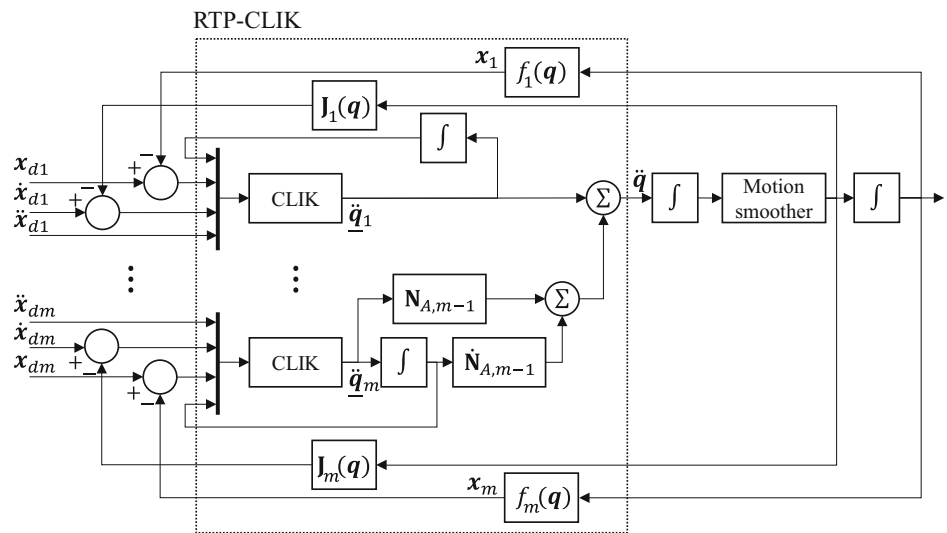
The block-diagram of the proposed approach including the RTP-CLIK with the motion smoother is sketched in the Fig. 4.

### 3 Jacobian coordinate transformation

Though a base frame ( $\mathcal{B}$ ) of the rescue robot is set on the pelvis like usual humanoid robots, task states and Jacobians for rescue motions using a whole-body have to be transformed with respect to other coordinates such as foot and horizon frames ( $\mathcal{F}, \mathcal{H}$ ). The each coordinate is expressed in the Fig. 5. In order to calculate the transformed Jacobian without a change of the base frame, it is possible to adopt a compatibility condition which means that the body center velocity induced from arbitrary limbs is identical, written as follows: (Kim et al. 2005)



**Fig. 4** Block-diagram of the proposed RTP-CLIK with the motion smoother.  $x_{di}, i \in (1, \dots, m)$  means a desired  $i$ -th task trajectory.  $f(q)$  represents forward kinematics



$$T_j(\dot{x}_{Lj} - J_j \dot{q}_j) = T_k(\dot{x}_{Lk} - J_k \dot{q}_k) \tag{16}$$

where

$$T_j = \begin{bmatrix} I_3 & [R_0 \ ^0r_j^\times] \\ \mathbf{0}_3 & I_3 \end{bmatrix} \in \mathbb{R}^{6 \times 6} \tag{17}$$

$\dot{x}_{Lj}$  denotes the  $j$ -th limb end velocity with regard to an arbitrary reference coordinate ( $\mathcal{W}$ ),  $R_0$  is the orientation matrix of body center with regard to a reference coordinate,  $^0r_j$  means the position vector from the body center to the  $j$ -th limb end with regard to the base frame.  $[\cdot]^\times$  is a skew-symmetric matrix for the cross product.

Using the compatibility condition, the Jacobians of the EE positions and COG with regard to a reference coordinate can be expressed as follows: (Kim et al. 2005; Sugihara and Nakamura 2002)

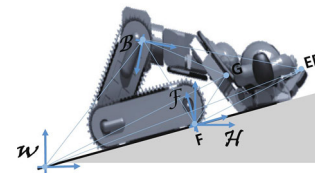
$$J_\Gamma = R_0 \left\{ \begin{matrix} {}^0J_\Gamma - {}^0J_F + \left[ ({}^0r_\Gamma - {}^0r_F)^\times \right] {}^0J_{\omega F} \end{matrix} \right\}, \tag{18}$$

$\Gamma \in \{G, EE\}$

where  $G, EE, F$  mean the positions of the COG, EE and supporting point fixed on the ground respectively. In this paper, the middle point of the both feet is regarded as the supporting point.  ${}^0J_F, {}^0J_{\omega F}$  represent the Jacobian of position and orientation of the supporting point.  ${}^0J$  denote a Jacobian with regard to the body frame. Also, the Jacobian of the EE orientation with regard to a reference coordinate can be derived with the identical procedure and then,

$$J_{\omega EE} = R_0 \left( {}^0J_{\omega EE} - {}^0J_{\omega F} \right) \tag{19}$$

Note that different reference coordinates are adopted for the EE and COG task;  $\mathcal{F}$  for the EE task and  $\mathcal{H}$  for the COG



**Fig. 5** Description of various coordinates

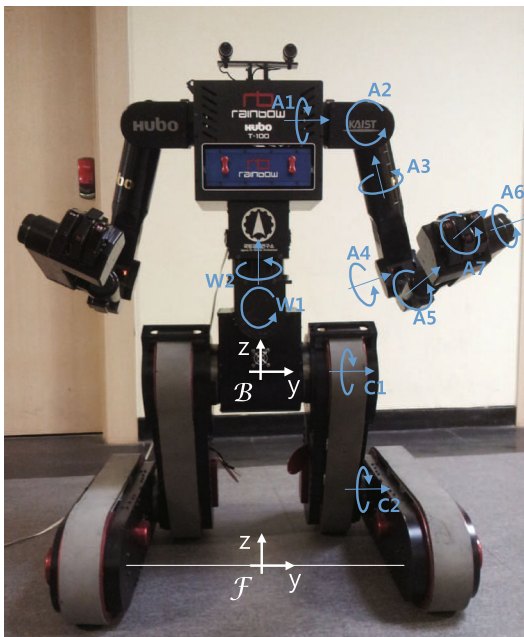
task. Terrain slope orientation which can be obtained from information of a IMU sensor and leg posture of the robot should be reflected on  $R_0$  when  $J_G$  is calculated for balancing problem. It is because the projected support polygon range and COG position can be changed in accordance with a terrain slope.

### 4 Numerical analysis and experiments

In this section, the performances between both the TP-CLIK and RTP-CLIK are compared numerically and then, experiments for the rescue motion applying the RTP-CLIK are tackled. We utilized a small-scaled simulator robot, Hubo-T100, in place of the real rescue robot. The Hubo-T100 was developed as a testbed for motion control verification, whose appearance is analogous to the rescue robot consisting of a dual-manipulator and a variable configuration platform as shown in the Fig. 6. Because it was spun off from a Hubo2+, whole size is smaller than the rescue robot so that its lifting capability is not substantial as much as the rescue robot can.

Trajectories for each task are generated as a point-to-point motion based on 5-th polynomials represented as Eq. (20) in order to regard acceleration constraints (Siciliano and Slotine 1991).

$$x_{di}(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{20}$$



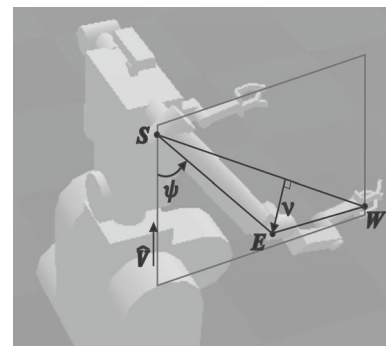
**Fig. 6** Description of a small-scaled simulator (Hubo-T100) with information on joint-axis and body/foot coordinates

The coefficients can be calculated by using the inverse problem as Eq. (21) with imposed initial ( $t_0$ ) and final ( $t_f$ ) positions corresponding motions and both velocity and acceleration set to zero.

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}^{-1} \begin{bmatrix} x_0 \\ 0 \\ 0 \\ x_f \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

**4.1 Numerical comparison between the both TP-CLIK and RTP-CLIK**

To analyze and compare the performance of between the both algorithms Eqs. (4) and (12) in terms of the algorithmic singularity robustness, numerical simulations for a special motion using only the upper-body of the robot were performed through a Webots dynamics software. In the simulations, the initial posture of the robot was set as the Fig. 8a. Also, a primary task was imposed to the EE pose and subtasks such as keeping the initial waist and elbow posture were subjected to avoid a self-motions. In order to induce the situation with the algorithmic singularity, a left arm motion towards left side was selected as a primary task, which will be conflicted to the subtasks. Here, a swivel angle was adopted to determine the elbow posture, which is defined by the angle between a vertical plane and the plane containing whole arm



**Fig. 7** Description of the swivel angle

as shown in the Fig. 7.  $W, E, S$  are wrist, elbow, shoulder position vectors, respectively.  $\hat{V}$  is a vertical unit vector and  $v$  represents the minimum distance from line  $SW$  to  $E$ . Using these vectors, the expression of the angle is as following: (Kreutz-Delgado 1990)

$$\psi = \text{atan2} \left( \hat{\eta}^T (\hat{V} \times v), \hat{V}^T v \right) \quad (22)$$

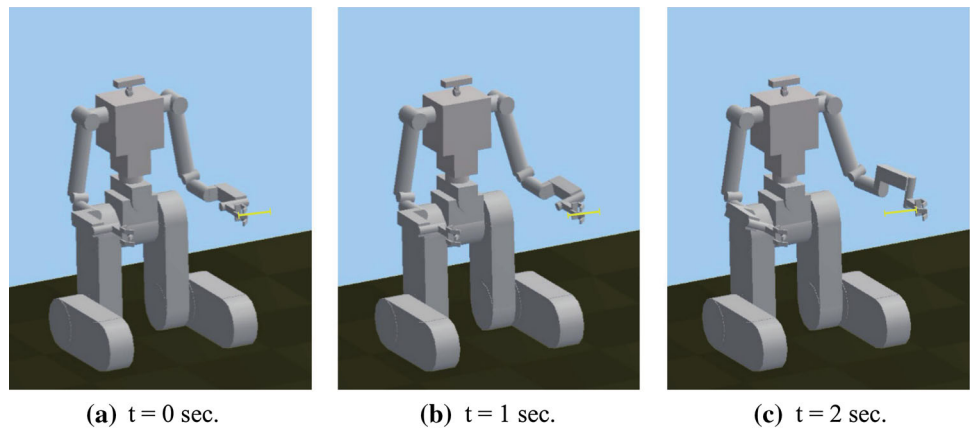
where  $\eta = W - S$  and  $[\hat{\cdot}]$  means a unit vector.

Table 1 shows the numerical results with maximum error of each task according to the algorithms. The TP-CLIK was combined with the DLS method with various perturbation values ( $\lambda^2 = 0.05, 0.07, 0.1$ ) for the algorithmic singularity. Firstly, let us compare the RTP-CLIK and TP-CLIK with  $\lambda^2 = 0.07$ . As shown in the table, the results of the EE position by the both algorithms seem to be satisfied with the enough small error. However, the performances of other tasks are clearly different according to the approaches. While the RTP-CLIK obtains better results in terms of the EE orientation (the error values denote the Euler angle), the TP-CLIK shows better performance of the subtasks. The Figs. 8 and 9 show the motion history of the robot in accordance with the TP-CLIK and RTP-CLIK, respectively. It seems that the TP-CLIK tries to keep the postures of waist and of left arm even with the twisted EE orientation but the RTP-CLIK intends to maintain the EE pose abandoning the postures of waist and of left arm. This is analogous tendency with the expectation that when the algorithmic singularity is happened, the RTP-CLIK tries to satisfy a primary task abandoning the performance of subtasks while the TP-CLIK with the DLS efforts for the all performances even distorting both a primary and subtasks. Surely, the deterioration of the primary task by TP-CLIK could be decreased by selecting bigger  $\lambda$ , but it tends to ignore the effort for the subtasks like the result by  $\lambda^2 = 0.1$  in the table. On the other hand, smaller  $\lambda$  could break the robustness of the algorithmic singularity. This can be confirmed from the result by  $\lambda^2 = 0.05$  in the table. Therefore, it is not easy to choose proper  $\lambda$  to hold the both task accuracy and robustness

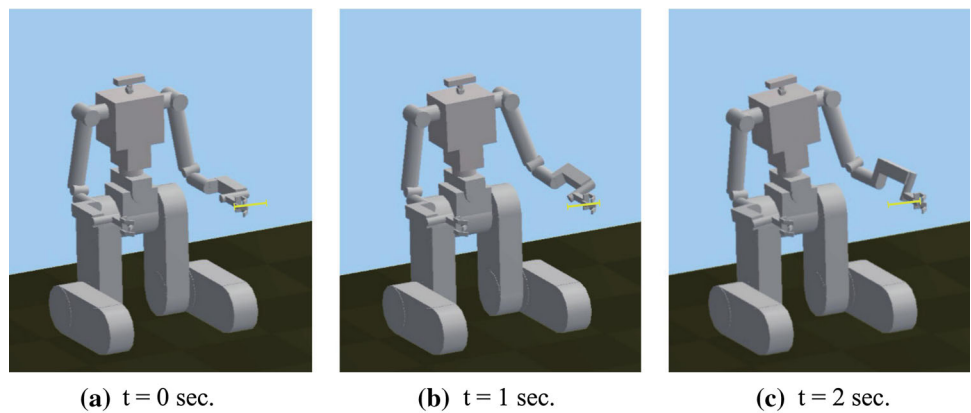
**Table 1** Maximum error of the both TP-CLIK with various  $\lambda^2$  and RTP-CLIK

Priority	Task	TP-CLIK ( $\lambda^2$ )			RTP-CLIK
		0.05	0.07	0.1	
1	EE position (m)	Divergence	2.8e−3	1.0e−3	1.7e−4
1	EE orientation (°)	Divergence	3.2e+0	4.2e−1	2.2e−1
2	Waist posture (°)	Divergence	3.3e+0	4.4e+0	5.2e+0
2	Swivel posture (°)	Divergence	5.7e+0	9.1e+0	1.3e+1

**Fig. 8** Motion history according to the left arm movement by the TP-CLIK with  $\lambda^2 = 0.07$ . Yellow line represents the desired trajectory of left EE. **a**  $t = 0$  s, **b**  $t = 1$  s, **c**  $t = 2$  s (Color figure online)



**Fig. 9** Motion history according to the left arm movement by the RTP-CLIK. **a**  $t = 0$  s, **b**  $t = 1$  s, **c**  $t = 2$  s



considering various motion scenarios, so that the RTP-CLIK is applied for the rescue robot in this paper.

**4.2 Experiments using a small-scaled simulator**

In this study, the rescue motions using whole-body DOFs are regarded for the experiments, which are made up lowering, forwarding, backwarding, contact, and lifting primitives sequentially. The contact phase which is accomplished by just elbow joint motion not using the IK is demanded to put a casualty on the manipulators. Also, the lifting primitive is operated through the joint motion too, in order to transform the robot posture to a pre-defined driving posture. As the simulator has quite redundant DOFs, further three subtasks besides a primary task were subjected to solve the IK for each primitives as shown in Table 2. Note that a hori-

zontal maneuver of the lower-arm as the task2 is requested specially for the rescue motions so that manipulators do not collide to the ground during lowering and forwarding phases. Therefore, a new projected elbow angle onto the  $x$ - $z$  plane of the  $\mathcal{F}$  frame was applied instead of the swivel angle. The balancing task was assigned with a relatively low priority, it is because the robot has the enough wide support polygon (0.4 m) to restrain the rollover when the COG is maintained in the vicinity of the center of the polygon even without exact tracking. Also, joint limit avoidance through the GPM whose performance index is written as Eq. (23) was considered to evade an unexpected leg posture as well as the physical joint restriction.

$$I(q) = -\frac{1}{2} \sum_{i=1}^n \left( \frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \tag{23}$$

**Table 2** Tasks assigned for rescue motion primitives

Priority	Task1	Task2	Task3	Task4
Task	EE pose	Waist posture and lower-arm level motion	COG balancing	Joint limit avoidance
Method	IK	IK	IK	GPM

The subscript  $M, m$  mean the maximum, minimum joint limit respectively and  $\bar{q}$  is the middle value of the range. Note that the rescue motions consisting of such many subtasks would have a conflict problem among the tasks, which induces the algorithmic singularity. We will confirm the RTP-CLIK’s ability to handle such a singularity problem from following experiments.

Trajectories for the EE pose and the COG were designed by using the Eq. (20). In the case of the orientation, although an initial and final values for the trajectory were set based on the Euler angle, the every desired values of the orientation were calculated after transforming into the quaternion formulation in order to avoid the well-known singularity problem of the Euler coordinate. For the COG task, the goal position of each phases is set at the projected center of support polygon (about  $-0.173$  m with respect to the  $\mathcal{H}$  coordinate shown in Fig. 5) except for the forwarding phase. The goal of the forwarding phase is differently positioned more ahead from the projected center in order to lessen the stretching motion of the upper body. The detailed description for each motion in the operational space is written in Table 3.

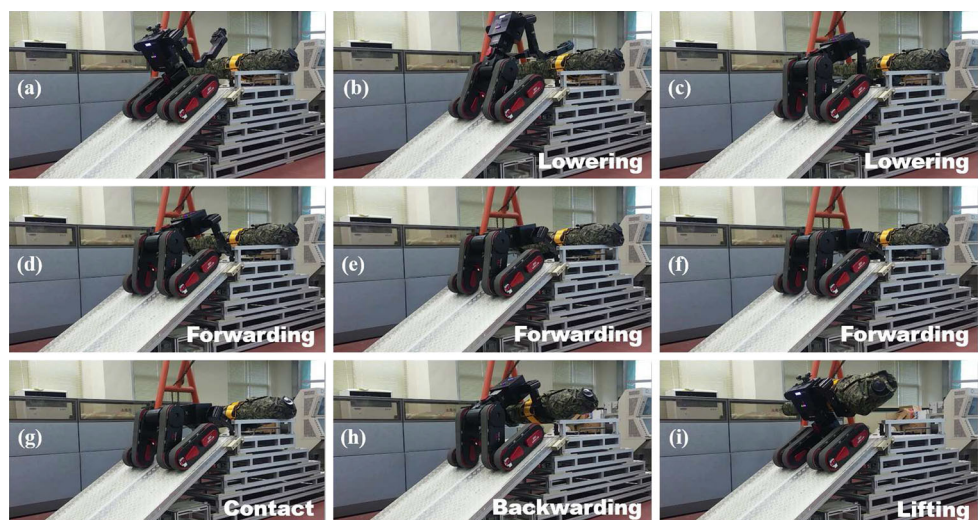
For the rescue scenario test, a dummy whose weight is about 15 kg in place of a real casualty is utilized and it was assumed to be laid horizontally with some gap between it and the ground in order to easily insert the EE into the gap and put him on the manipulators. Here, because the forwarding motion has to be operated on the level ground, trajectories of EE pose are designed in the  $\mathcal{H}$  coordinate like the COG task. Note that how to handle a casualty laid directly on the ground is out of scope of this paper. Also, it was supposed that the robot is on the  $30^\circ$  slope in order to clearly confirm the balancing ability.

Figure 10 shows the consecutive snapshots of the rescue motions consisting of the five primitives sequentially. The forwarding phases are described more concretely because the behavior during the phase is more significant. It seems that the rescue mission for the laid dummy was carried out properly, trying to maintain the lower-arm level motion. In addition, the robot kept its balance with regard to the  $\mathcal{H}$  frame considering the slope environment even under the extremely stretching condition of a upper body. The resultant behavior seems to be not complicated, but it does not with regard to the

**Table 3** Task description for rescue motions in the operational space ( $30^\circ$  slope)

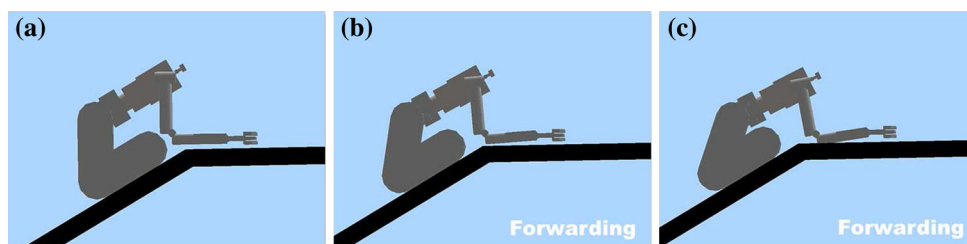
(Priority) Task	Coordinate	Lowering	Forwarding	Backwarding
(1) EE position (x, y, z)	$\mathcal{H}$	(0, 0, $-0.2$ ) m	(0.4, 0, 0) m	( $-0.3$ , 0, 0.2) m
(1) EE orientation	Hand	Initial orientation	Initial orientation	Initial orientation
(2) Waist posture	$\mathcal{B}$	Initial posture	Initial posture	Initial posture
(2) Elbow posture	$\mathcal{H}$	$0^\circ$	$0^\circ$	–
(3) COG (x)	$\mathcal{H}$	$-0.173$ m	$-0.13$ m	$-0.173$ m

**Fig. 10** Consecutive snapshots of the experiment for the proposed rescue motions ( $30^\circ$  slope). Each snapshot is arrayed in alphabetical order

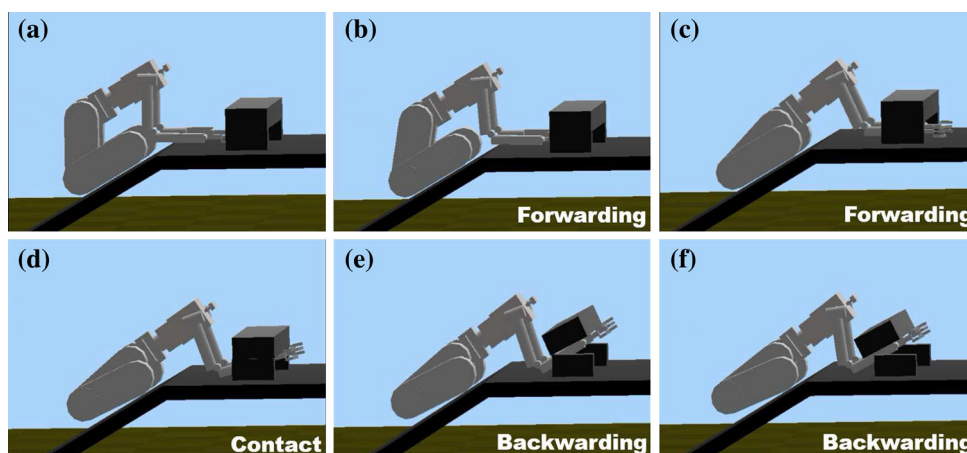




**Fig. 11** Consecutive snapshots of the numerical simulation for only Task1 (EE pose) without the all subtasks

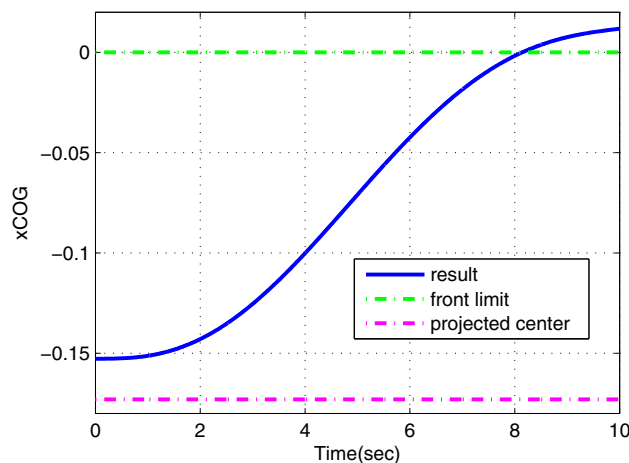


**Fig. 12** Consecutive snapshots of the numerical simulation not considering the balancing task (30° slope)



IK problem considering the task-priority approach to handle the algorithmic singularity among four tasks. Therefore, we need to analyze effect of the task-priority and singularity robustness stage by stage.

Firstly, let us investigate the Task2 to keep the lower-arm level motion. If the lower-arm level motion is not reflected, the lower-arm would be possible to collide with the ground during the forwarding phase though the robot could keep the desired EE pose due to the redundant DOFs, as shown in Fig. 11. Secondly, if the balancing task is not considered for the rescue motion on the slope, the robot would overturn at the moment of contact with the dummy because the COG could not be located in the projected support polygon any more. To confirm this expectation, numerical simulation through the Webots was performed and the snapshots of the results are expressed in Fig. 12. From the figure, we can see that the whole body motion during the forwarding phase is quite different from the previous result considering the balancing task. Since the lower body behaviors more ahead due to no constraint on the balancing, it can be easily expected that the COG of the robot becomes more ahead too. This can be checked from Fig. 13 in detail. As approaching, the COG is getting closer to the front limit (0m means the front end of the track), and then it is slightly beyond the limit finally. Consequentially, at the moment to contact the dummy, it is confirmed that the robot becomes to be tipped over. Although the robot does not seem to overturn perfectly thanks to the arm in the numerical simulation, such situation could cause a critical situation in real experiment because

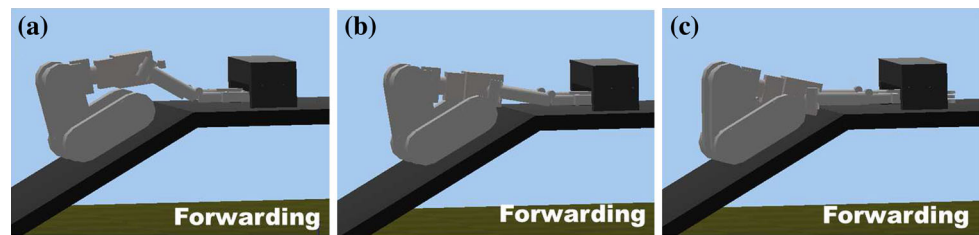


**Fig. 13** Results of the projected COG position during the forwarding phase in the simulation without the balancing task

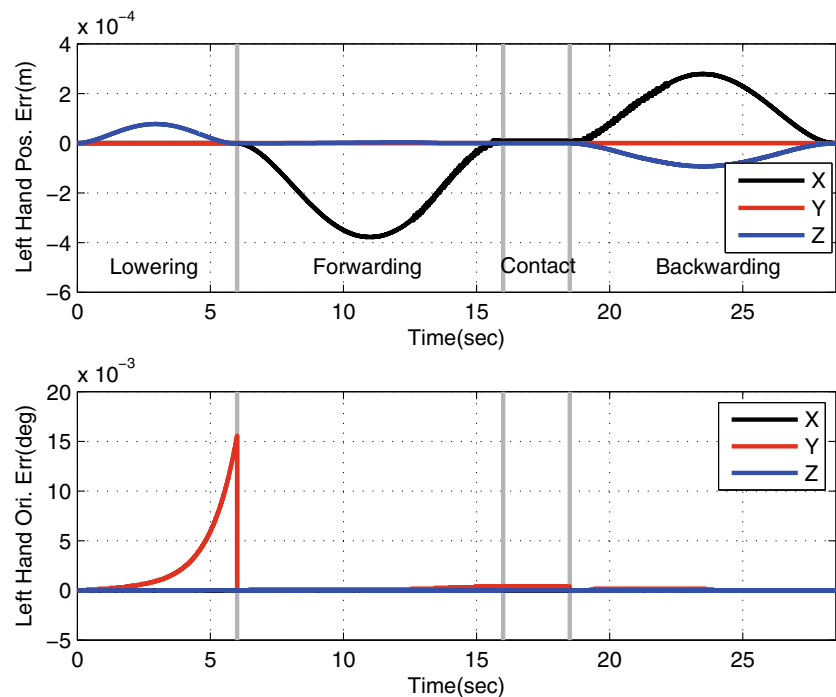
the contact area between the track of the lower body and slope becomes too small to prevent the robot from slipping on the slope. Also, if the joint limit avoidance to evade an unexpected leg posture and joint restriction is not applied, it would make undesirable motion to induce a risky situation. Such situation is confirmed from Fig. 14 which shows collision between the upper body and ground when the Task4 is ignored. Consequentially, we can conclude that the RTP-CLIK generated good performance making an effort to reflect the complicated whole tasks.

As noted previously, however, tracking of such subtasks could not be satisfied perfectly since they are not independent

**Fig. 14** Numerical simulation result in the case of ignoring the Task4 (to evade an unexpected leg posture as well as the physical joint restriction)



**Fig. 15** Results of error of left EE position and orientation. Solid gray line is expressed to differentiate each phase

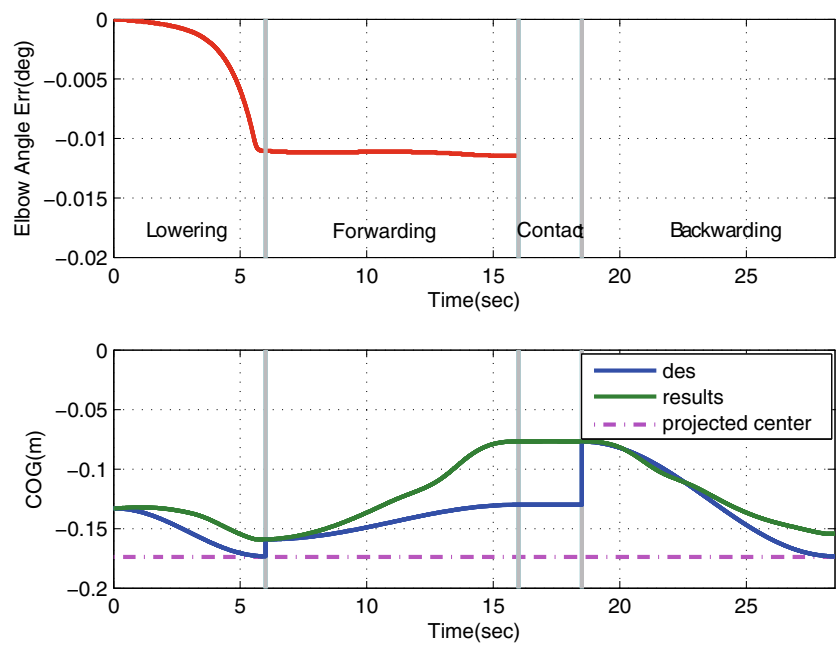


of the primary task, which means the algorithmic singularity. Such prediction can be confirmed obviously by the graphs from the Figs. 15 and 16. Since the dual-manipulator maneuvers symmetrically during the rescue motion, only the results of the left arm are represented in the figures. Also, the error of the left elbow posture is not described after the end of the forwarding phase (16s) because it is not demanded during the phases left. The pose of the EE was perfectly tracked regardless of the algorithmic singularity due to its primary priority (Task1). Also, it seems that the performance of the elbow level motion as the Task2 was satisfied with small error because the redundant domain from the Task1 is enough for the Task2. However, the result of the Task3 (balancing) is worse than the higher tasks. It means that there is conflict between the balancing task and higher task, so the redundancy from the higher tasks (Task1, 2) is not enough to work for the Task3. Especially, we can see the error of the balancing task during the forwarding phase becomes worse. It is because the desired position of EE (forward to 0.4 m) is too far to maintain the assigned COG trajectory. Surely, it would be better if the COG goal during the phase is repositioned more ahead as long as possible, but the robot could be tipped

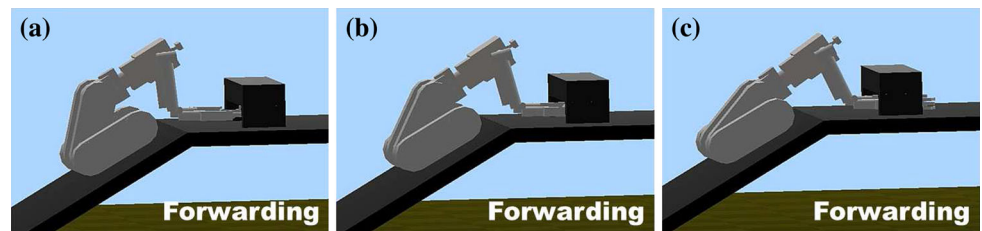
over at the moment of contact with the dummy. For arbitrary missions, therefore, it is better to set the desired COG position on the vicinity of the center of the support polygon. Although there are some tracking error of the COG, it is not significant for the balancing thanks to the wide support polygon. From the experiment results, we can conclude that the supposed algorithm has the ability to handle the algorithmic singularity reflecting the task-priority in the case of existence of some conflict among tasks.

On the other hand, if the TP-CLIK with the DLS approach is applied for this scenario, it is not easy to select a proper  $\lambda$  because we cannot anticipate how much conflict between the upper tasks and balancing task is. As mentioned previously, too small value could induce divergent solutions or too large value could ignore the subtasks, and such expectation can be confirmed from Figs. 17 and 18. In Fig. 17, we can see that the result of the simulation with big  $\lambda$  ( $\lambda = 3.0$ ) seems to be similar with motions of Fig. 12 which shows the results not considering the balancing task, so we can conclude that the balancing task was ignored due to the big  $\lambda$ . On the other hand, Fig. 18 shows divergent results by small  $\lambda$  ( $\lambda = 0.1$ ), even though the  $\lambda$  is not pretty small value. It means that the

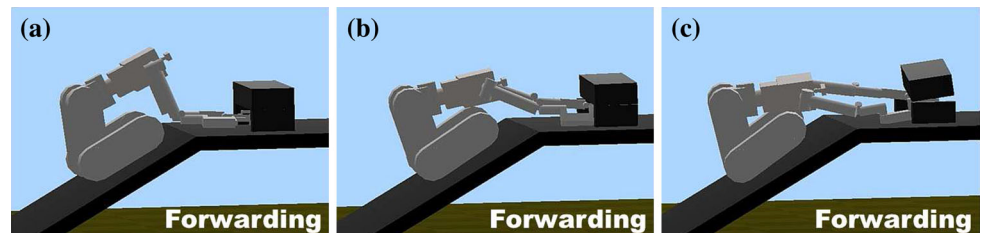
**Fig. 16** Results of left elbow posture error and COG position. The COG position is expressed with regard to  $\mathcal{H}$  coordinate. Dotted purple line means the projected center of the support polygon onto  $\mathcal{H}$  coordinate (Color figure online)



**Fig. 17** Results when the TP-CLIK with DLS (big  $\lambda : 3.0$ ) is applied



**Fig. 18** Results when the TP-CLIK with DLS (small  $\lambda : 0.1$ ) is applied



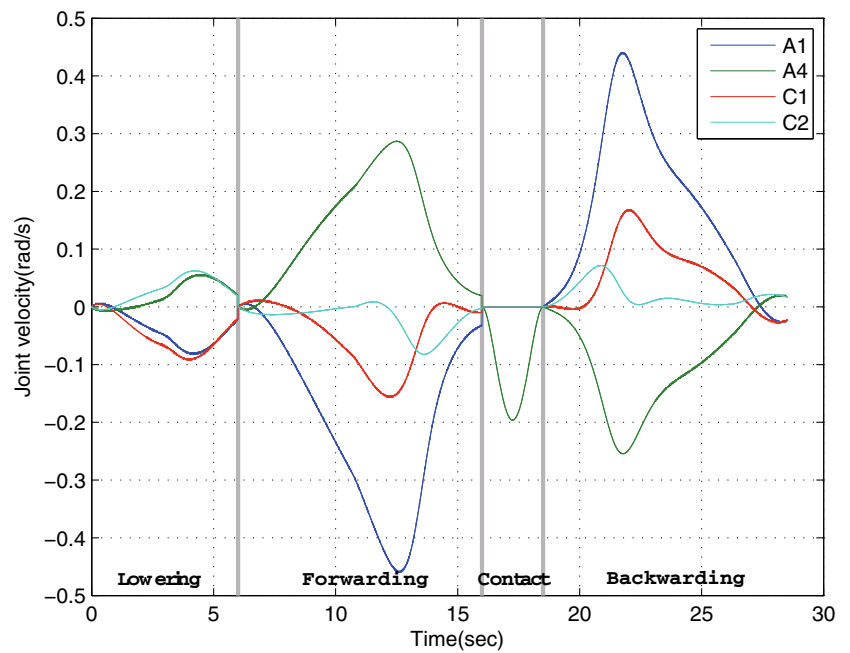
amount of algorithmic singularity of that situation is quite significant, so it is not possible to satisfy whole tasks simultaneously. In order to operate the robot for various missions, therefore, the proposed approach is more safe and effective than the TP-CLIK.

Figure 19 shows the results of four joint velocities corresponding of A1, A4, C1, C2. Other joint velocities are omitted due to too small value. As shown in the figure, there are undesirable end-motions causing non-zero joint velocities at the end of each phase, which are the anticipated phenomenon mentioned in the Sect. 2.2. To handle the phenomenon, the suggested motion smoother can be applied with  $\Delta = 0.5$  s. Figure 20 describes the results of joint velocities adopting

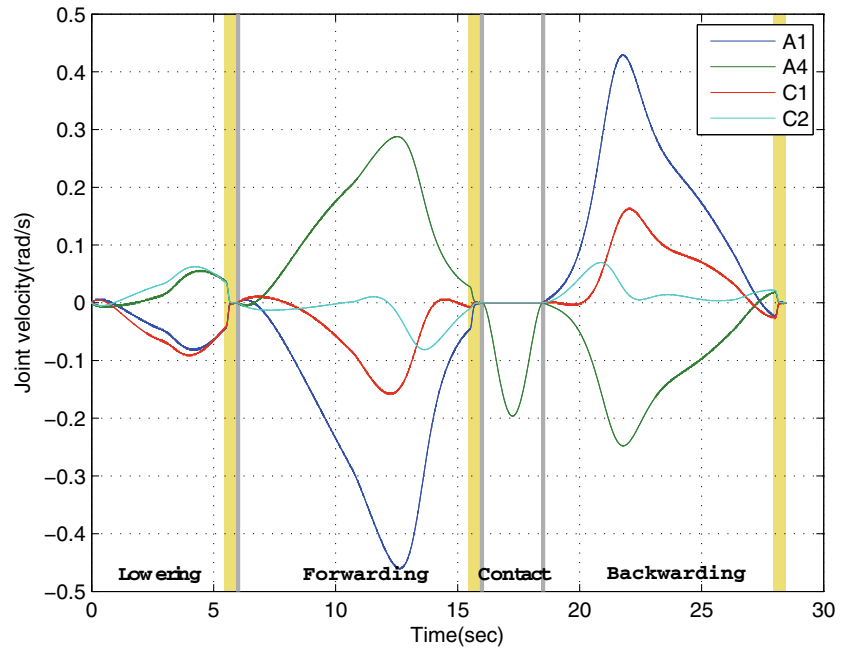
the motion smoother and it is found that the joint velocities converge to zero at the end of each phase.

Finally, we prepared another experiment with more extreme and dangerous scenario to highlight the balancing problem. We set the robot and dummy on the identical plane of the  $-30^\circ$  slope. In this case, the robot cannot stretch his arms as much as the previous experiment because the COG can be easily beyond the front limit of the support polygon according to the forwarding motion. However, since the robot and dummy are on the identical slope unlike the situation of the previous test, the robot can approach the dummy by driving. So, the robot position for the forwarding phase can be closer to the dummy thanks to the driving. Table 4

**Fig. 19** Joint velocities of A1, A4, C1, C2 without the motion smoother



**Fig. 20** Joint velocities of A1, A4, C1, C2 with the motion smoother. Yellow region represents the interval ( $\Delta$ ) of each phase for the motion smoother (Color figure online)



**Table 4** Task description for rescue motions in the operational space ( $-30^\circ$  slope)

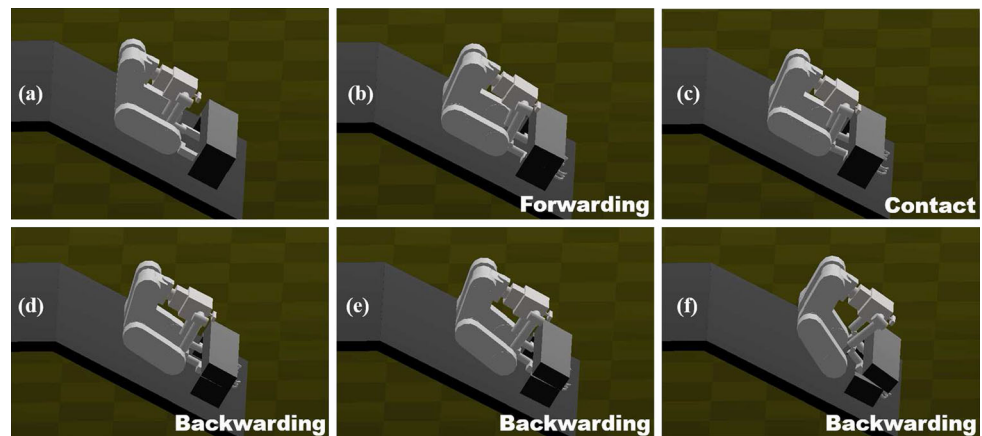
(Priority) Task	Coordinate	Forwarding	Backwarding
(1) EE position (x, y, z)	$\mathcal{F}$	(0.2, 0, 0) m	(-0.15, 0, 0.2) m
(1) EE orientation	Hand	Initial orientation	Initial orientation
(2) Waist posture	$\mathcal{B}$	Initial posture	Initial posture
(2) Elbow posture	$\mathcal{F}$	$0^\circ$	-
(3) COG (x)	$\mathcal{H}$	-0.173 m	-0.26 m

shows a detailed task description for the rescue motions in the operational space. Note that the coordinate of the tasks for the EE position and Elbow posture is changed due to the

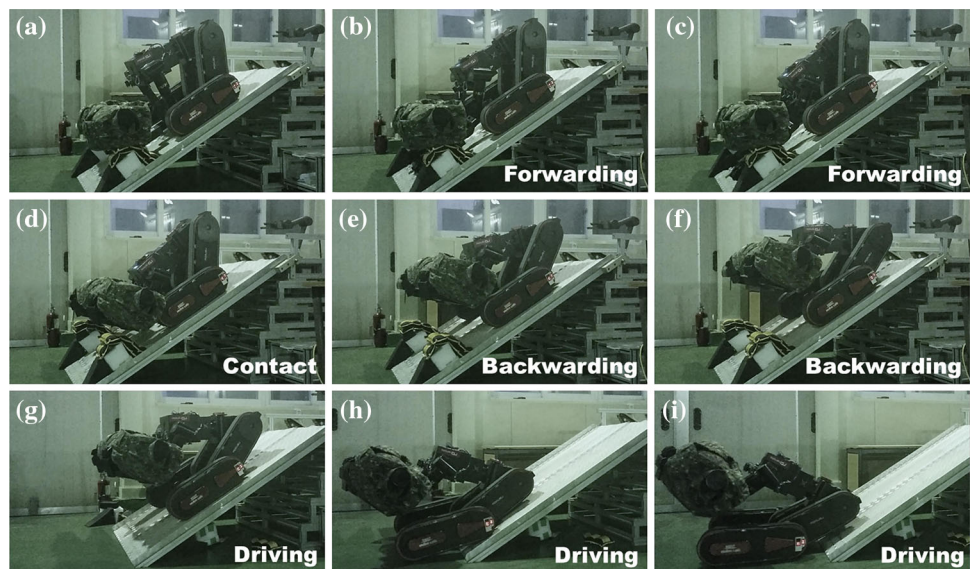
modified situation. Also, the desired COG position during the backwarding phase is set behind the center of the support polygon in order to secure enough margin for the balancing.



**Fig. 21** Consecutive snapshots of the numerical simulation not considering the balancing task ( $-30^\circ$  slope). Each snapshot is arrayed in alphabetical order



**Fig. 22** Consecutive snapshots of the numerical simulation reflecting the balancing task ( $-30^\circ$  slope)



Figures 21 and 22 describe the consecutive snapshots of the numerical simulation without the balancing task and of the experimental result reflecting the balancing effort, respectively. From Fig. 21, we can see that the lower body maneuvers towards according to the forwarding motion, inducing the forward movement of the COG, and then it makes the robot become to be tipped over after the contact phase. On the other hand, as shown in Fig. 22, when the balancing task is reflected, it seems that the robot tries to move the COG backwards as far as possible to satisfy the desired COG trajectory. Thanks to the motion, the robot can safely lift the dummy without a rollover situation. Besides, we can see stable driving of the robot lifting the dummy in the figure.

## 5 Conclusions

This paper introduced a research related with a rescue motion using a whole-body of a rescue robot consisting of dual-manipulator and variable configuration mobile platform.

Since the rescue motion to lift a casualty from the ground has to be limited due to the characteristics of the robot configuration and balancing issue, a specific motion strategy was suggested for the robot. In order to effectively carry out the motion, some subtasks were taken into account thanks to redundant DOFs and the IK problem was solved by a CLIK approach with task-priority strategy for real-time operation. Because the original approach of the TP-CLIK has an algorithmic singularity problem, a RTP-CLIK was adopted with modified formulations of 2nd-order equations for smooth motion. Besides, a motion smoother was suggested to eliminate the unnatural end-motion induced by the RTP-CLIK. The pros and cons of the RTP-CLIK against the TP-CLIK was studied analytically and numerically and it was concluded that the RTP-CLIK is more acceptable for any motion in accordance with various missions. Also, in order to reflect different target coordinates in accordance with tasks, the Jacobian coordinate transformation based on the compatibility condition was considered without changing the base frame. A small-scaled simulator as the testbed of experiments

was introduced and the rescue motions of the simulator were operated well enough on the 30°/−30° slope by the proposed algorithm. However, this study did not include an in-depth discussion of how to put a casualty on the manipulators. In the rescue experiments, a dummy instead of a casualty was assumed to be laid with gap from the ground. Therefore, it will be further work to safely handle and lift a casualty laid on the ground in any condition.

**Acknowledgements** This research was supported by a grant for the Project managed by the Agency for Defense Development. “Technology development for a rescue robot capable of lifting over 120 kgf”, funded by the Dual-use technology program.

**Appendix: Analysis on algorithmic singularity**

For easy analysis on the algorithmic singularity issue, 1st-order equation of the TP-CLIK consisting of only two tasks is handled instead of the Eq. (4) and the equation is rewritten as follows:

$$\begin{aligned} \dot{q} &= \dot{q}_h + \tilde{J}_l^\dagger (\dot{x}_l - J_l \dot{q}_h) \\ \dot{q}_i &= J_i^\dagger \dot{x}_i, \quad i \in h, l \\ N &= (I_n - J_h^\dagger J_h) \end{aligned} \tag{24}$$

To compare both algorithms with regard to the algorithmic singularity, some terms associated with the singularity are available to be reexpressed based on the singular value decomposition (SVD). For the Eq. (24), let us define the decomposition of the  $J_h$  and  $J_l$  in the form

$$J_h = \sum_{i=1}^r \sigma_i u_i v_i^T \tag{25}$$

$$J_l = \sum_{i=1}^{n-r} \sigma_{l,i} u_{l,i} v_{l,i}^T \tag{26}$$

$J_h$  is assumed to be full-rank for simplification. Then, decomposition of  $\tilde{J}_l$  is induced as follows: (Chiaverini 1997)

$$\begin{aligned} \tilde{J}_l &= U \Xi V \\ &= (u_{l,1} \dots u_{l,n-r}) \\ &\quad \times \begin{pmatrix} \sigma_{l,1} v_{l,1}^T v_{r+1} & \dots & \sigma_{l,1} v_{l,1}^T v_n \\ \vdots & \ddots & \vdots \\ \sigma_{l,n-r} v_{l,n-r}^T v_{r+1} & \dots & \sigma_{l,n-r} v_{l,n-r}^T v_n \end{pmatrix} \\ &\quad \times \begin{pmatrix} v_{r+1}^T \\ \vdots \\ v_n^T \end{pmatrix} \end{aligned} \tag{27}$$

It is noted that a loss of rank of  $\tilde{J}_l$  is related with  $\Xi$ , which is caused by the kinematic singularity of subtask ( $\sigma_{l,n-r} = 0$ ) or the algorithmic singularity ( $v_{l,i}^T v_j = 0, i \in [1, n-r], j \in [r+1, n]$ ). Also, as the inner products get closer to zero, inversion of the ill-conditioned  $\tilde{J}_l$  makes null space velocity excessively to maintain tracking accuracy of subtask.

On the other hand, inversion term of the Eq. (11) as the RTP-CLIK can be written as follows: (Chiaverini 1997)

$$(I_n - J_h^\dagger J_h) J_l^\dagger = \sum_{i=1}^{n-r} \sum_{j=r+1}^n \frac{v_{l,i}^T v_j}{\sigma_{l,i}} v_j u_{l,i}^T \tag{28}$$

While the kinematic singularity ( $\sigma_{l,i}$ ) of the subtask can lead joint divergence, there is no effect by the algorithmic singularity. This is because  $(I_n - J_h^\dagger J_h) J_l^\dagger$  makes the null space velocity of subtask decrease when the associated term,  $v_{l,i}^T v_j$ , gets closer to zero.

**References**

Antonelli, G. (2009). Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5), 985–994.

Baerlocher, P., & Boulic, R. (1998). Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proceedings of international conference on intelligent robotics and systems, Victoria, Canada*.

Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3), 398–410.

Chiaverini, S., Egeland, O., & Kaneström, R. (1991). Achieving user-defined accuracy with damped least-squares inverse kinematics. In *Proceedings of international conference on advanced robotics, Pisa, Italy*.

Kajita, S., Kanehiro, F., Kaneko, K., & Fujiwara, K. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of IEEE international conference on robotics and automation, Taipei, Taiwan*.

Kanoun, O., Lamiroux, F., & Wieber, P. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27, 785–792.

Kim, D., Choi, Y., Oh, Y., & You, B. (2005). Real-time generation of humanoid motion with the motion-embedded COG Jacobian. In *Proceedings of international conference on control, automation and systems, Kintex, Korea*.

Kim, I., & Oh, J. (2013). Inverse kinematic control of humanoids under joint constraints. *International Journal of Advanced Robotic Systems*, 10(74), 1–12.

Kreutz-Delgado, K. (1990). Kinematic analysis of 7 DOF anthropomorphic arms. In *Proceedings of IEEE international conference on robotics and automation, Cincinnati, Ohio*.

Maciejewski, A. A., & Klein, C. A. (1988). Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, 5(6), 527–552.

Mitsi, S., Bouzakis, K.-D., & Mansour, G. (1995). Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joint limits. *Mechanism and Machine Theory*, 30(5), 653–663.

- Nakamura, Y. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3), 163–171.
- Nakamura, Y., Hanafusa, H., & Yosikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *International of Robotics Research*, 6(2), 3–15.
- Sardain, P., & Bessonnet, G. (2004). Forces acting on a biped robot. Center of pressure-zero-moment point. *IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 34(5), 630–637.
- Sciavicco, L., & Siciliano, B. (1988). A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal of Robotics and Automation*, 4(4), 403–410.
- Siciliano, B. (1990a). A closed-loop inverse kinematic scheme for on-line joint-based robot control. *Robotica*, 8(3), 231–243.
- Siciliano, B. (1990b). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3), 201–212.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modeling, Planning and Control*. Berlin: Springer.
- Siciliano, B., & Slotine, J. E. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Proceedings of international conference on advanced robotics, Pisa, Italy*.
- Sugihara, T., & Nakamura, Y. (2002). Whole-body cooperative balancing of humanoid robot using COG Jacobian. In *Proceedings of international conference on intelligent robots and systems, Lausanne, Switzerland*.
- Vukobratovic, M., & Borovac, B. (2004). Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1), 157–173.
- Wang, L., & Chen, C. (1991). A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transaction on Robotics and Automation*, 7(4), 489–499.
- Yoshikawa, T. (1984). Analysis and control of robot manipulators with redundancy. In *Proceedings of the first international symposium on robotic research, Cambridge, MA*.



**Wonsuk Lee** received the B.S. degree in aerospace engineering from Korea Aerospace University, Goyang, Korea, in 2005, and M.S. and Ph.D. degrees in aerospace engineering from KAIST, Daejeon, Korea, in 2007 and 2011, respectively. From 2011 to 2012, he was a Postdoctoral Researcher with the KAIST Aerospace Systems and Control Laboratory. Since 2013, he has been a Senior Researcher with Defense Unmanned Technology Directorate, Agency for Defense

Development, Daejeon. His research interests are concerned with UAV control, target tracking, cooperative robot control, and whole-body motion control.



**Youngwoo Lee** received the B.S. and M.S. degree in electrical engineering from KAIST, Daejeon, Korea, in 2011 and 2013, respectively. Since 2013, he has been a Researcher with Defense Unmanned Technology Directorate, Agency for Defense Development, Daejeon. His research interests are concerned with robot software architecture, real-time systems, and real-time networks.



Dynamic balance stabilization.

**Gyuhyun Park** received the B.S. degree in Mechanical and Aerospace engineering from Seoul National University, Seoul, Korea in 2012, and M.S. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, Korea, in 2014. Since 2014, he has been a Researcher with Defense Unmanned Technology Directorate, Agency for Defense Development, Daejeon. His research interests are concerned with Whole-body motion control and



2006, he joined the Agency for Defense Development, Daejeon, Korea, where he is Senior researcher. His current research interests include control and dynamical systems, robotics, estimation, and target tracking.

**Seongil Hong** received the B.S. degree in mechanical engineering from Pusan National University, Pusan, Korea, in 2002, and the M.S. degree in mechanical engineering from Pohang University of Science and Technology, Pohang, Korea, in 2004. From 2004 to 2006, he was with the research and development center of Hyundai-Kia Motor Company, Whasung, Korea, where he was engaged in noise, vibration, and harshness problems of engines and vehicles. In



**Younsik Kang** received the B.S. degree in mechanical engineering from Pusan National University, Pusan, Korea, in 1984, and M.S. and Ph.D. degrees in mechanical engineering from KAIST, Daejeon, Korea, in 1986 and 1997, respectively. Since 1986, he has been a Principal Researcher with Defense Unmanned Technology Directorate, Agency for Defense Development, Daejeon. His research interests are concerned with tracking and stab-

lization for Electro Optical Tracking System, control algorithm, sensor signal processing and mechanical/electronic stabilization system design.