Kinematic Parameter Calibration for Humanoid Robot Using Relative Pose Measurement in Walking Motion

Jee-eun Lee[1] and Jaeheung Park[1]

Abstract— **Robots in use often need to be re-calibrated, due to the mechanical part deformation resulted from unexpected collisions. However, current calibration algorithms are not easy to be applied to humanoids and they require complex experiment setup. This is because humanoids have difficulties in gathering absolute pose measurement since they do not have a fixed frame on its body. On the other hand, the relative pose over time can be easily measured by a camera or a lidar that are usually attached to the humanoids' head. In this paper, we present a convenient and low-cost calibration method for humanoids by using relative pose measurements in walking motion. Our framework starts from the assumption that the sole of the support foot at every single footstep can be thought of as the temporary fixed frame while walking. Then, based on the decision of the supporting foot, we model the forward kinematics of an open chain from the foot to the head. In the description of a kinematic model, we use the local product of exponentials (POE) formula to take advantages of its simplicity and continuity in computation. Then, we solve the least square problem formulated to minimize the error between the actual measurement and nominal value computed from the corresponding joint angle data. Lastly, simulation results will be discussed to verify the feasibility of this framework.**

I. INTRODUCTION

In general, the problems in robotics such as motion planning, control, and even mapping unknown environment are highly dependent on the model and its parameters. The kinematics of robots is an example, of which we often set the parameters based on its design value. However, the actual values may not be the same as the CAD model because there is so much chance of error possibly caused from manufacturing, assembly tolerance and part deformation. The inaccuracy in model generated from those errors sometimes causes serious problems; for example, humanoids with inaccurate kinematics can walk in a biased direction, and even it could fail to localize itself when its mapping algorithm uses unreliable odometry calculated from inaccurate kinematics.

Many proposed methods to calibrate robots such as [1], [2], [3], [4] are developed to be suitable for industrial manipulators. These methods require the measurements of end-effector pose from the base frame in various configuration, and these measurements are not easy to be taken for humanoids because humanoids do not have any fixed frame on their body. In [5], a camera on the robot's head was used to measure the pose of the sole frame from the head. They hold the robot's back in place and attach a checkerboard to its foot, so that a camera can detect it and measure the poses. Another method proposed in [6] also measures the foot position while a robot is lying down. They used an industrial robot and captured the foot poses by connecting the robot's tool to the humanoid's sole.

However, re-calibration is often required for humanoids after a collision, which occurs frequently and causes deformation in robot geometry. The problem is that the methods mentioned above are not easily available but require specific experiment setup to fix the robot's back to the ground. To address this problem, the method to calibrate a robot while a robot is walking was suggested based on the assumption that a robot will keep both feet on the same plane during the double support phases[7]. However, robots in a double support phase usually have unstable contacts.

In this paper, we propose a new calibration framework to enable a humanoid to be efficiently calibrated through walking motion. Instead of using absolute pose measurements, we take advantages of using time sequential relative pose data. It can be easily taken from 3D Visual Odometry (VO) or Lidar Odometry (LO) that are widely used in Simultaneous Localization and Mapping (SLAM) algorithms. In addition, it does not require any additional setup so that we can get this data while a robot is just walking.

II. PROBLEM FORMULATION

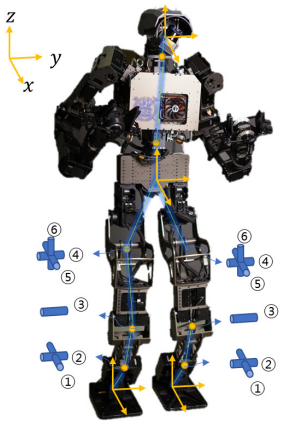
This framework consists of three steps: (i) Decision of a Supporting foot, (ii) Data processing for calibration, and (iii) Formulation of the optimization problem including setting an appropriate forward kinematics and error model that should be minimized to obtain proper kinematic parameters.

A. Decision of a Supporting Foot

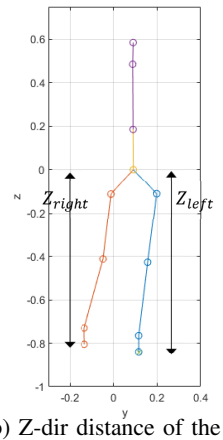
The proposed calibration framework for a humanoid system is built based on the assumption that the support foot can be considered as a temporary fixed frame while a robot is walking. Generally, a humanoid holds a ground contact continuously changing from single to double support foot phase. Considering the fact that the contact during double support foot phase tends to be more unstable, we will only use the data that is supposed to be in a single support foot phase. Sensor attached to the foot(e.g. a force torque sensor or a pressure sensor) can be used to determine which foot is the support foot, but in this paper, we will just compare the z-direction distance of the left and right foot.

*This work was supported by Industrial Strategic Technology Development Program (No. 10077538) funded by the Ministry of Trade, Industry Energy (MI, Korea) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF2015R1A2A1A10055798).

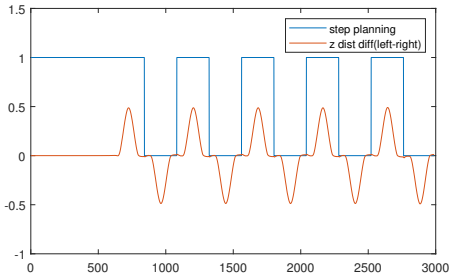
[1]are with the DYROS (Dynamic Robotics Systems) Lab, Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Republic of Korea. Jaeheung Park is the corresponding author. {tina0303, park73}@snu.ac.kr



(a) DYROS-JET Configuration



(b) Z-dir distance of the foot



(c) Step planning and z-dir difference between left and right foot

Fig. 1: Description of DYROS-JET foot and walking motion

Fig. 1 shows the difference of z-direction distances between the left and right foot when the robot is walking. Z-directional distance from the sole to the torso of the robot can be calculated from nominal kinematic parameters of the robot. Then we can decide a supporting foot with the threshold value ϵ as:

$$\text{Supporting Foot} = \begin{cases} \text{Left}, & \text{if } Z_{left} - Z_{right} > \epsilon \\ \text{Right}, & \text{if } Z_{left} - Z_{right} < -\epsilon \end{cases} \tag{1}$$

$\epsilon = 0.05$ will be used for our simulation later.

B. Synchronization of Data

One of the major problems with this calibration framework is asynchronous data captured by two different kinds of equipment: the joint angle data measured from the encoder and the relative pose data from the camera. In addition, some humanoids system might have separate computers to process each data efficiently. In this case, we can use Network Time Protocol (NTP) software such as OpenNTPD[8] or chrony[9], to synchronize the clocks between computer systems. Then, we will get each data with the synchronized time stamp that represents the observation time.

Let $(t_i, \mathbf{q}_i)$ denote the pair of observation time and joint angle data that are usually obtained at high frequency(200Hz-1kHz), and (t'_j, I_j) denote the pair of observation time and image frame data that are obtained at relatively low frequency(15-30Hz). Since the relative pose will be used in this framework, we just need to take this pair in the form of $(\mathbf{q}_k, \mathbf{q}_{k+1}, dg_{k,k+1})$, where $dg \in SE(3)$ refers to the relative pose estimated from the consecutive images (I_k, I_{k+1}).

In implementation of this process, cubic spline interpolation, or smoothing interpolation algorithms seem to be a good possible way to achieve it. If the joint angle data is smooth and reliable enough, then cubic spline that guarantees the second derivative continuity would be applicable. Otherwise, we should use another interpolation method like kernel smoothing functions that prevent us from overshoot problems caused by the noise. Let $f : t \in \mathbb{R} \rightarrow \mathbf{q} \in \mathbb{R}^n$ be the mapping function of the joint angle with respect to the observation time, which can be obtained by interpolation algorithms that are mentioned above.

$$f(t) = f(t_1, ...t_N, \mathbf{q}_1, ..., \mathbf{q}_N) \tag{2}$$

Then, we could construct the data pairs $(\mathbf{q}_k, \mathbf{q}_{k+1}, dg_{k,k+1})$ from observed data for this calibration framework as:

$$\begin{aligned} \mathbf{q}_k &= f(t'_k) \\ \mathbf{q}_{k+1} &= f(t'_{k+1}) \\ dg_{k,k+1} &= dg(I_k, I_{k+1}) \end{aligned} \tag{3}$$

C. Forward Kinematics Model

A generic forward kinematics for an n-DOF serial robot can be modeled via minimal Product-of-Exponential (POE) formula as:

$$g(\mathbf{q}) = e^{[\hat{\zeta}_1]q_1} e^{[\hat{\zeta}_2]q_2} \cdots e^{[\hat{\zeta}_n]q_n} M \tag{4}$$

where q_i is the joint variable, $\hat{\zeta}_i$ is a unit screw axis of the joint i expressed in the fixed frame O_s, and lastly, $M = g(0) \in SE(3)$ is the end-effector configuration when all joints are set to zero. Here, we used $\hat{\cdot}$ to indicate that the screw is a unit screw axis, and $[(\cdot)]$ represents the skew-symmetric matrix of the vector $(\cdot)$. These representations are similar to the ones given in [10].

Without using any link reference frame, the kinematic parameters can have a unique solution in minimal POE formula by describing all the screws and end-effector configuration in the base frame coordinate. However, this formula does not identify each link's geometry and therefore it is not convenient to work with Unified Robot Description Format (URDF)[11], a widely used format to describes kinematics. For these reasons, we decided to identify the parameters based on the link reference frame to enhance the applicability of this framework.

By separating zero end-effector configuration M into $e^{[\xi_1]}e^{[\xi_2]} \cdots e^{[\xi_n]}e^{[\xi_{tcp}]}$, we can describe a unit screw axis in the link reference frame, and model the forward kinematics solely for the link geometric information. This formulation was firstly introduced in [12] as the local POE formula.

$$g(\mathbf{q}) = e^{[\xi_1]}e^{[\hat{s}_1]q_1} e^{[\xi_2]}e^{[\hat{s}_2]q_2} \cdots e^{[\xi_n]}e^{[\hat{s}_n]q_n} e^{[\xi_{tcp}]} \tag{5}$$

where,

$$[\xi_i] = \begin{bmatrix} [\omega_i] & v_i \\ 0 & 0 \end{bmatrix} \in se(3) \tag{6}$$

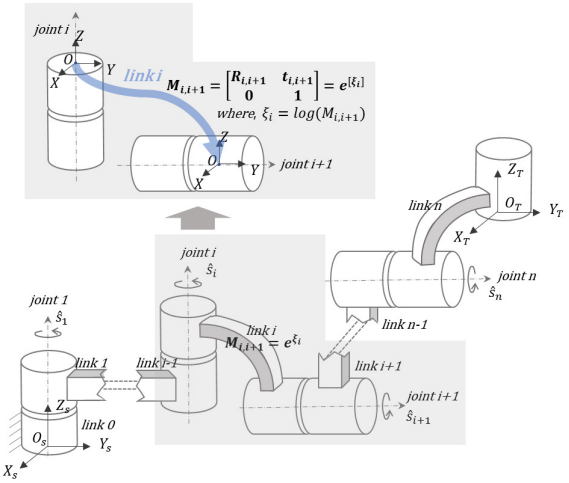


Fig. 2: Forward kinematics of a n-DOF serial robot

with

$$v_i, \omega_i \in \mathbb{R}^3,$$
$$[\omega_i] = \begin{bmatrix} 0 & -\omega_{3i} & \omega_{2i} \\ \omega_{3i} & 0 & -\omega_{1i} \\ -\omega_{2i} & \omega_{1i} & 0 \end{bmatrix} \in so(3) \tag{7}$$

In the equation (5), $[\xi_i] \in se(3)$ represents the twist of transformation matrix between the link reference frames and each $[\hat{s}_i] \in se(3)$ represents the joint i's unit screw axis that is expressed in the link reference frame, which we will arbitrarily set in advance. Therefore, we can fix $\hat{s}_i$ and the only variables of the problem become ξ_i.

D. Linearized Kinematic Error Model

1) Absolute Pose Error Model: The linearized kinematic error model using absolute pose is successfully derived in [13], which can be obtained by linearizing the forward kinematic equation (5) as follows

$$\delta g g^{-1} = \big(\frac{\partial g}{\partial \xi}\delta\xi\big) g^{-1} \tag{8}$$

As shown in [13], redundant parameters should not be involved in the error model at the same time. Therefore, it is sufficient to consider the only variable ξ_i to get derivative of g.

Now, let g_a and g_n be the configuration of actual and nominal end-effector frames, respectively, where g_a is obtained from the measurement data and g_n is computed based on the nominal parameter values. Then $\delta g g^{-1} = (g_a - g_n)g_n^{-1} = g_a g_n^{-1} - I$. With the first-order approximation of exponential taylor expansion, $g_a g_n^{-1} = I + log(g_a g_n^{-1}) + log(g_a g_n^{-1})^2/2! + \cdots$, $\delta g g^{-1} \in se(3)$ is also given by [14], [15].

$$\delta g g^{-1} = g_a g_n^{-1} - I \approx log(g_a g_n^{-1}), \tag{9}$$

Hence, after completing the procedure of pose measurements, the identification of kinematic parameters is to solve the least-square problem

$$\min \| log(g_a g_n^{-1}) - \big(\frac{\partial g}{\partial \xi}\delta\xi\big) g^{-1} \|^2 \tag{10}$$

2) Relative Pose Error Model: Similarly, we can derive kinematic error model subject to the relative pose. From the equation (5), relative pose difference between two poses determined by joint variables $\mathbf{q}^{t_1}$ and $\mathbf{q}^{t_2}$ can be written as follows:

$$\begin{aligned} dg(\mathbf{q}^{t_1}, \mathbf{q}^{t_2}) &= g(\mathbf{q}^{t_1})^{-1} g(\mathbf{q}^{t_2}) \\ &= e^{-[\xi_{tcp}]} e^{-[\hat{s}_n] q_n^{t_1}} e^{-[\xi_n]} \cdots e^{-[\hat{s}_2] q_2^{t_1}} e^{-[\xi_2]} e^{-[\hat{s}_1] q_1^{t_1}} \\ &\quad \cancel{e^{-[\xi_1]}} \cdot \cancel{e^{[\xi_1]}} e^{[\hat{s}_1] q_1^{t_2}} \cdots e^{[\xi_n]} e^{[\hat{s}_n] q_n^{t_2}} e^{[\xi_{tcp}]} \end{aligned} \tag{11}$$

As you can see from the equation, relative pose is not dependent on the configuration of the first joint, so it is impossible to identify the parameters of the first joint if we use relative pose in calibration. However, even when we use absolute pose data in calibration, the geometry of the first joint from the soles of the foot is hardly identified because a camera is only possible to measure the pose of checkerboard attached to the foot rather than the sole itself. Thus, we will not consider this seriously. Then, the error model for the relative pose is given by:

$$\delta dg(\mathbf{q}^{t_1}, \mathbf{q}^{t_2}) \cdot dg(\mathbf{q}^{t_1}, \mathbf{q}^{t_2})^{-1} = \big(\frac{\partial dg}{\partial \xi}\delta\xi\big) dg^{-1} \tag{12}$$

which refers to the difference of pose change δdg with respect to the different twist $\delta\xi$ in the frame of later end-effector configuration $g(\mathbf{q}^{t_2})$. And then, for the simple notation, let ξ_{n+1} denote ξ_{tcp} and, T_k, T'_k represent:

$$T_k(\mathbf{q}) = e^{[\xi_1]} e^{[\hat{s}_i] q_1} \cdots e^{[\xi_k]} e^{[\hat{s}_k] q_k} = \prod_{i=1}^{k} e^{[\xi_i]} e^{[\hat{s}_i] q_i} \tag{13}$$

$$\begin{aligned} T'_k(\mathbf{q}) &= e^{-[\xi_{tcp}]} e^{-[\hat{s}_n] q_n} e^{-[\xi_n]} \cdots e^{-[\xi_{k+1}]} e^{-[\hat{s}_k] q_k} \\ &= g(\mathbf{q})^{-1} e^{[\xi_1]} \prod_{i=1}^{k-1} e^{[\hat{s}_i] q_i} e^{[\xi_{i+1}]} \end{aligned} \tag{14}$$

Then, the equation (12) is derived as

$$\begin{aligned} \big[\delta dg \cdot dg^{-1}\big]^V &= \big[\delta e^{-[\xi_{tcp}]} e^{[\xi_{tcp}]}\big]^V + \cdots \\ &+ \big[T'_2(\mathbf{q}^{t_1})\big(\delta e^{-[\xi_2]} e^{[\xi_2]}\big) T'_2(\mathbf{q}^{t_1})^{-1}\big]^V \\ &+ \big[g(\mathbf{q}^{t_1})^{-1} T_1(\mathbf{q}^{t_2})\big(\delta e^{[\xi_2]} e^{-[\xi_2]}\big) T_1(\mathbf{q}^{t_2})^{-1} g(\mathbf{q}^{t_1})\big]^V + \cdots \\ &+ \big[g(\mathbf{q}^{t_1})^{-1} T_n(\mathbf{q}^{t_2})\big(\delta e^{[\xi_{tcp}]} e^{-[\xi_{tcp}]}\big) T_n(\mathbf{q}^{t_2})^{-1} g(\mathbf{q}^{t_1})\big]^V \end{aligned} \tag{15}$$

By introducing the adjoint map and derivation of $\big[\delta e^{[\xi_i]} e^{-[\xi_i]}\big]^V$ given in [13], equation (15) can now be derived in fully explicit form as:

$$\begin{aligned} \big[\delta dg \cdot dg^{-1}\big]^V &= \sum_{i=1}^{n} \Big(Ad_{g(\mathbf{q}^{t_1})^{-1} T_i(\mathbf{q}^{t_2})} \big[\delta e^{[\xi_i]} e^{-[\xi_i]}\big]^V \\ &\quad + Ad_{T'_{i+1}(\mathbf{q}^{t_1})} \big[\delta e^{-[\xi_i]} e^{[\xi_i]}\big]^V \Big) \\ &= \sum_{i=1}^{n} \Big(X_i A_i \delta\xi_i + Y_i B_i \delta\xi_i \Big) \end{aligned} \tag{16}$$

where,

$$\begin{aligned}\left[\delta e^{[\xi_i]}e^{-[\xi_i]}\right]^V =&\Bigg(\mathbf{I}_6+\frac{4-\theta_i\sin(\theta_i)-4\cos(\theta_i)}{2\|\omega_i\|^2}\Omega_i\\&+\frac{4\theta_i-5\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^3}\Omega_i^2\\&+\frac{2-\theta_i\sin(\theta_i)-2\cos(\theta_i)}{2\|\omega_i\|^4}\Omega_i^3\\&+\frac{2\theta_i-3\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^5}\Omega_i^4\Bigg)\delta\xi_i\\=&A_i\delta\xi_i \qquad (17)\end{aligned}$$

$$\begin{aligned}\left[\delta e^{-[\xi_i]}e^{[\xi_i]}\right]^V =&\Bigg(\mathbf{I}_6+\frac{4-\theta_i\sin(\theta_i)-4\cos(\theta_i)}{2\|\omega_i\|^2}\Omega_i\\&-\frac{4\theta_i-5\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^3}\Omega_i^2\\&+\frac{2-\theta_i\sin(\theta_i)-2\cos(\theta_i)}{2\|\omega_i\|^4}\Omega_i^3\\&-\frac{2\theta_i-3\sin(\theta_i)+\theta_i\cos(\theta_i)}{2\|\omega_i\|^5}\Omega_i^4\Bigg)\delta\xi_i\\=&B_i\delta\xi_i \qquad (18)\end{aligned}$$

$$Ad_{g(\mathbf{q}^{t_1})^{-1}T_i(\mathbf{q}^{t_2})} = X_i \qquad (19)$$

$$Ad_{T'_{i+1}(\mathbf{q}^{t_1})} = Y_i \qquad (20)$$

and

$$\Omega_i = \begin{bmatrix}[\omega_i] & 0\\ [v_i] & [\omega_i]\end{bmatrix} \qquad (21)$$

$$\theta_i = \|\omega_i\| = (\omega_{1i}^2+\omega_{2i}^2+\omega_{3i}^2)^{1/2} \qquad (22)$$

With the actual relative pose dg_a measured during the movement and nominal value dg_n computed accordingly using kinematic parameters, $\delta dg\cdot dg^{-1}$ can also be expressed as

$$\begin{aligned}\left[\delta dg\cdot dg^{-1}\right]^V &= [(dg_a-dg_n)\cdot dg_n^{-1}]^V\\&= [dg_a dg_n^{-1}-I]^V\\&\approx \log(dg_a dg_n^{-1}) \qquad (23)\end{aligned}$$

E. A Least Square Problem

With the observed data on various configurations while a robot is walking, the least square problem to find optimal kinematic parameters can be formulated from the equation (16) and (23) as

$$\min\sum_{j=1}^{m}\|\log(dg_a dg_n^{-1})_j-\sum_{i=1}^{n}\left(X_iA_i+Y_iB_i\right)_j\delta\xi_i\|^2 \qquad (24)$$

where $(\cdot)_j$ denotes jth observation data of m observations in total. Then, we can obtain the solution by solving the least square problem.

$$\begin{aligned}\Delta x^* &= \arg\min_{\Delta x}\left\|\begin{bmatrix}\Delta y_1\\ \Delta y_2\\ \vdots\\ \Delta y_m\end{bmatrix}-\begin{bmatrix}J_1\\ J_2\\ \vdots\\ J_m\end{bmatrix}\Delta x\right\|^2\\&= \arg\min_{\Delta x}\left(\Delta y-J\Delta x\right)^T\left(\Delta y-J\Delta x\right)\\&= (J^TJ)^{-1}J^T\Delta y \qquad (25)\end{aligned}$$

with

$$\begin{aligned}\Delta y_j &= \log(dg_a dg_n^{-1})_j\in\mathbb{R}^6\\ J_j &= \sum_{i=1}^{n}\left(X_iA_i+Y_iB_i\right)_j\in\mathbb{R}^{6\times(6n+6)}\\ \Delta x &= \left(\delta\xi_1^T,\ldots,\delta\xi_n^T\right)^T\in\mathbb{R}^{6n+6}\\ \Delta y &\in \mathbb{R}^{6m}\\ J &\in \mathbb{R}^{6m\times(6n+6)} \qquad (26)\end{aligned}$$

Finally, we can obtain optimal kinematic parameters ξ by updating screws at each iteration until the error converges to the value close to zero.

$$\xi_i^* = \xi_i+\delta\xi_i \qquad (27)$$

III. SIMULATION RESULTS AND DISCUSSION

In this section, simulation experiments are presented for our calibration with DYROS-JET described in Fig. 1a. More specific information on DYROS-JET can be found in [16]. The overall simulation process is described in Fig. 3.

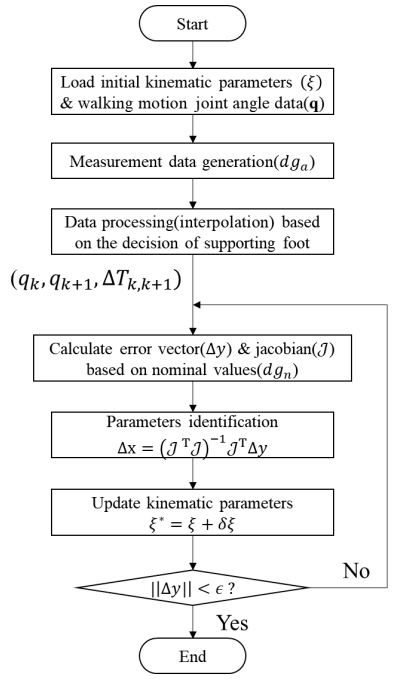
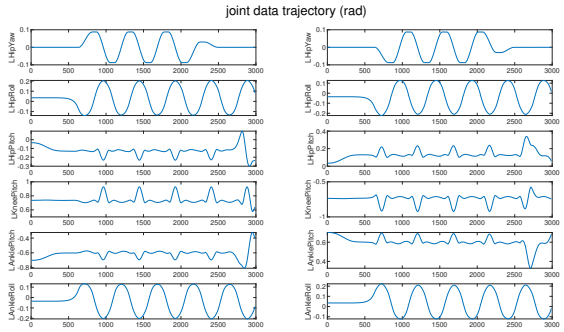
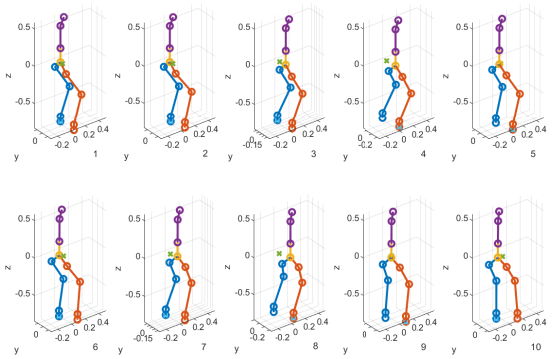


Fig. 3: simulation process diagram



(a) Joint trajectory of the robot lower-body



(b) Snapshots of the walking motion

Fig. 4: Joint trajectory and configurations of the humanoid robot walking along the circle

A. Experiment Setup

1) Motion Data: The motion of a robot that walks along the circle for about 15 seconds is used in this experiments. If we use motion data that does not move a particular joint between any two links, the information on the two links can not be identified. Thus, we used circular walking motion that moves all joints, of which the joint trajectory and the configuration sequence are shown in Fig 4. For similar reason, we also assumed that the upper-body of a humanoid as one link.

2) Kinematic Parameters: To evaluate the performance of our kinematics calibration algorithm, the actual kinematic parameters are virtually derived from the nominal parameters, which are presented in TABLE I. Then, we generate virtual measurement along the joint trajectory described in Fig. 4 based on the kinematics model with actual parameters.

3) Virutal Measurement: To realize our simulation closer to the real environment, we assumed that the joint trajectory is given at 200Hz and the relative pose measurement is given at 30Hz. Then, we can obtain 80 data pairs $(\mathbf{q}_{t_1}, \mathbf{q}_{t_2}, dg_a(t_1, t_2))$ of left foot and 68 data pairs of right foot by applying the supporting foot criterion defined in section II-A to the walking motion data(15s). In addition, we perform the experiment in two ways to verify the feasibility of this framework in practice: without the measurement noise and with the measurement noise that is uniformly distributed.

	nominal link screw(ξ_n)	actual link screw(ξ_a)	unit screw($\hat{s}$)
ω_1	0,0,0	-0.02,0.01,0	-1,0,0
v_1	0,0.075	0.0096,-0.0057, 0.0750	0,0,0
ω_2	0,0,0	0.0200,0,-0.0100	0,1,0
v_2	0,0,0	0.0050,0, 0.0050	0,0,0
ω_3	0,0,0	0,0,0	0,1,0
v_3	-0.06,0,0.368	-0.0700, 0.0010, 0.3780	0,0,0
ω_4	0,0,0	0, 0.0200, 0.0100	0,1,0
v_4	-0.152,0,0.339	-0.1449,-0.0002, 0.3475	0,0,0
ω_5	0,0,0	0, 0.0200, 0.0010	-1,0,0
v_5	0,0,0	0.0010,-0.0100, 0.0010	0,0,0
ω_6	0,0,0	0, 0.0010,-0.0200	0,0,1
v_6	0,0,0	0.0100,-0.0010, 0.0100	0,0,0
ω_{tcp}	0,0,0.524	0.0052,-0.0195, 0.5336	
v_{tcp}	0.021,-0.116,0.698	0.0378,-0.1130,0.7078	

TABLE I: The local POE kinematic parameters of the left foot of DYROS-JET

link screw	calibrated without noise	calibrated with noise
ω_1	0,0,0	0,0,0
v_1	0,0, 0.0750	0,0,0.0750
ω_2	0, 0.0183,-0.0100	-0,0.0179,0.0005
v_2	0,-0.0023, 0.0050	0,0.0008,-0.0185
ω_3	0,0,0	-0.0018,0,-0.0017
v_3	-0.0769,-0.0023, 0.3766	-0.0817,0.0008,0.3879
ω_4	-0.0001,0, 0.0100	-0.001,0.0003,0.01000
v_4	-0.1477,-0.0023, 0.3462	-0.1465,0.0008,0.3518
ω_5	0, 0.0216, 0.0010	-0.0083,0.021,-0.0105
v_5	0.0055,-0.0023, 0.001	-0.0102,0.0007,0.0061
ω_6	0, 0.0010,-0.0045	-0.0083,0.0041,-0.0027
v_6	0.0055,-0.0011, 0.0099	-0.0102,0.0243,0.0013
ω_7	0.0051,-0.0196, 0.5181	0.0031,-0.0195,0.5200
v_7	0.0369,-0.1132, 0.7078	0.0378,-0.1154,0.6991

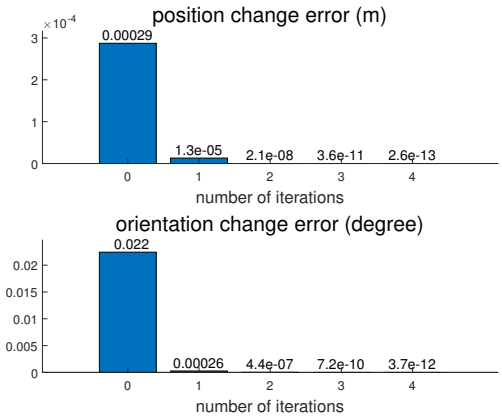
TABLE II: The local POE kinematic parameters obtained after calibration

The level of noise is set in the range [-0.2 0.2](in millimeter) and [-0.1 0.1](in milli-radian) and applied as a form of twist like $dg_{a,noise} = dg_a(q_{t_1}, q_{t_2}) \cdot e^{\xi_e}$, with uniformly distibuted noise ξ_e. We determined the noise size based on the recent results of the performance of stereo camera Visual Odometry (VO), which has translation error $0.65 \sim 2.6\%$ and orientation error $0.0014 \sim 0.0078$ deg/m, with the average of relative pose used in our work, about 9mm/s.

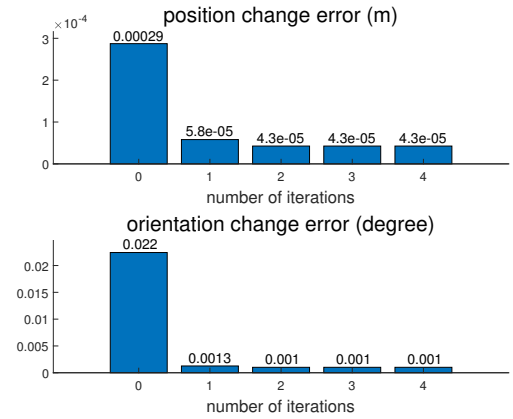
B. Simulation Result and Discussion

As we discussed in the previous section, the parameters of local POE kinematics cannot be identified with the uniqueness but just estimated near the initial value. Thus, we need to use the proper tricks to deal with the null space in jacobian inverse to solve the least square problem described in the equation (25). The most general method to solve this problem is to add damped term prior to taking inverse of the matrix.

$$\Delta x* = (J^T J + \sigma D)^{-1} J^T \Delta y \tag{28}$$



(a) without noise



(b) with noise

Fig. 5: Relative pose error mean during the iteration of calibration

Where, σ is a value small enough, and D is a diagonal matrix. If we want to compensate the error uniformly on the link, we can set D to an identity matrix, or if we want to compensate the error proportional to the link length, we can benefit from using a diagonal matrix of which terms are inversely proportional to the link length. In this paper, we used an identity matrix for D, and $0.01 \cdot \lambda^{+}_{min}(J^T J)$ for σ. λ^{+}_{min} is the smallest eigenvalue of the matrix that is larger than 0 or an equivalent constant(we used $1e{-}8$ for this). Then, calibrated link screws are obtained as given in Table II.

The relative position and orientation errors during each iteration of the calibration with and without the noise are illustrated in Fig. 5 respectively. As shown in the figure, an average error in relative position and orientation rapidly decrease by 15% and 4.6% respectively, even when we apply the measurement noise term.

IV. CONCLUSIONS

A new low-cost and convenient method to calibrate the kinematic parameters of a humanoid has been proposed. Our calibration framework is developed to use measurements during walking motion based on the assumption that the support foot at each single footstep can be considered as the temporary fixed frame while a robot is walking. In addition, we derived the kinematic error model with respect to the relative pose(pose change over time). Relative pose measurement can be easily obtained from visual odometry algorithms with cameras that are generally attached to the robot head. Then, we carried out simulations to estimate the kinematic parameters from the proposed calibration method. From the simulations, we verified that our calibration method can reduce the relative pose error effectively. All things considered, since the proposed method has high applicability and does not require additional high-cost experiment setup, we expect this algorithm will take a lot of advantages in a humanoid calibration.

REFERENCES

[1] L. Everett, M. Driels, and B. Mooring, "Kinematic modelling for robot calibration," in *IEEE International Conference on Robotics and Automation*, vol. 4, 1987, Conference Proceedings, pp. 183–189.

[2] Z. Roth, B. Mooring, and B. Ravani, "An overview of robot calibration," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.

[3] W. Khalil, S. Besnard, and P. Lemoine, "Comparison study of the geometric parameters calibration methods," *International Journal of Robotics Automation*, vol. 15, no. 2, pp. pp. 56–67, 2000.

[4] J.-Q. Xuan and S.-H. Xu, "Review on kinematics calibration technology of serial robots," *International journal of precision engineering manufacturing*, vol. 15, no. 8, pp. 1759–1774, 2014.

[5] T. Kastner, T. Röfer, and T. Laue, "Automatic robot calibration for the nao," in *Robot Soccer World Cup*. Springer, 2014, pp. 233–244.

[6] R. Khusainov, A. Klimchik, and E. Magid, "Humanoid robot kinematic calibration using industrial manipulator," in *International Conference on Mechanical, System and Control Engineering (ICMSC)*. IEEE, 2017, Conference Proceedings, pp. 184–189.

[7] F. Liu and L. Tang, "The calibration method of humanoid robot based on double support constraints," in *International Conference on Intelligent Robotics and Applications*. Springer, 2017, pp. 185–192.

[8] "Openntpd: a free, easy to use implementation of the network time protocol," Avilable:://www.openntpd.org/index.html, [Accessed: 2019-02-19].

[9] "chrony: a versatile implementation of the network time protocol (ntp)," Avilable:https://chrony.tuxfamily.org/, [Accessed: 2019-02-19].

[10] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[11] "Urdf: Unified robot description format," Available:http://wiki.ros.org/urdf, [Accessed: 2019-02-19].

[12] I.-M. Chen, G. Yang, C. T. Tan, and S. H. Yeo, "Local POE model for robot kinematic calibration," *Mechanism and Machine Theory*, vol. 36, no. 11-12, pp. 1215–1239, nov 2001.

[13] R. He, Y. Zhao, S. Yang, and S. Yang, "Kinematic-parameter identification for serial-robot calibration based on poe formula," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 411–423, 2010.

[14] K. Okamura and F. C. Park, "Kinematic calibration using the product of exponentials formula," *Robotica*, vol. 14, no. 4, pp. 415–421, 1996.

[15] F. C. Park and K. Okamura, *Kinematic calibration and the product of exponentials formula*. Springer, 1994, pp. 119–128.

[16] S. Kim, M. Kim, J. Lee, S. Hwang, J. Chae, B. Park, H. Cho, J. Sim, J. Jung, H. Lee, *et al.*, "Team snu's control strategies for enhancing a robot's capability: Lessons from the 2015 darpa robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 2, pp. 359–380, 2017.