2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

An On-line Gravity Estimation Method using Inverse Gravity Regressor for Robot Manipulator Control

Joonhee Jo[1,2], DongHyun Lee[1], Duc Trong Tran[3], Yonghwan Oh[1] and Sang-Rok Oh[1]

Abstract— **When a robotic manipulator is controlled, computing gravity force of the robot is the primary issue. Exact model parameters are not easy to be known in the practical robot system due to the uncertainty of the robot dynamics. Hence, the gravity force is presented by a combination of gravity regressor and robot dynamic parameters and is compensated by the estimation of uncertain robot dynamic parameters. Previous researches conducted estimation by using transpose of gravity regressor and full form of dynamic parameters which is not general form however this paper estimates the gravity force using the generalized gravity regressor which is regardless of the dimension and structure of the robot under the quasi-static state. Once the estimation is completed, the estimated value can be used to compute the gravitational force and control the robot. It is shown that the generalized decomposition of gravity regressor and estimation process. The results are validated through an experiment by implementing the algorithm on an upper-body dual arm robot.**

I. INTRODUCTION

In the beginning of the robotics research, the control of robot manipulator has studied by trying to solve the dynamics of the system. Computing the control torque using Lagrange equation [1] is not easy for the redundant and multi-manipulators which have many degrees of freedom (DoFs). Newton-Euler method is another method to compute the torque however its computation time issue comes up, and both has a uncertainty in the robot system. Instead, many robots are controlled using the proportional-integral-derivative (PID) control, virtual spring damper together with friction, gravity compensation, hybrid control, and so on. Researches in [2], [3] assume that the robot parameters such as link mass, length, center of mass (CoM) position, and kinematic information are exactly known however it is not sure in the practical robot system due to uncertainties caused by manufacturing errors or additional weight by wires, especially friction. Those effects are significant in controlling industrial robots when the robot needs to perform accurate and repeatable tasks. Hence, the estimation of the robot model parameters are necessary to obtain more accurate information of the system and to control the robot appropriately.

Estimation researches have proceeded and been classified by uncertainty on robot actual model, dynamic parameters, and so on. Sensory feedback is necessary for each method to estimating the parameters such as encoder, vision sensor. The estimation of gravity by uncertain dynamic parameters are researched [4], [5], [6], [7], [8] by Arimoto et. al, and uncertain gravity regressor [9], [10] with various applications. Cheah et. al. researched on uncertain jacobian [11] together with estimation of dynamic parameters [12], [13], [14], [15], [16] and gravity regressor [17]. Some further researches are studied for contact force estimation and hybrid control [18], [19], [20].

Those researches on gravity estimation by using transpose of gravity regressor have no general form and always have to use full form of dynamic parameter estimation. Hence, this paper tries a new method to generalize the gravity decomposition to estimate the gravity force regardless of the dimension and structure of the robot. Once generalized, the decomposed gravity regressor and dynamic parameters are not necessary to be derived again. The estimated gravity using the generalized form is implemented to the upper-body dual arm robot simulator and compared with the control of robot with and without gravity.

The paper presents the gravity decomposition process in II through the example of three link manipulator and implementation to the simulation in III. Finally, discussion and conclusion are denoted in IV.

II. DYNAMICS AND KINEMATICS OF THE ROBOT

The paper considers a manipulator which have all revolute joints. The dynamics of the robot manipulator which has n DoFs is expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

where $\boldsymbol{q} = [q_1, \ldots, q_n]^T \in \Re^n$ is the joint angles of the robot, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the angular velocity and acceleration. In addition, $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{n\times n}$ is the inertia matrix of the system which is symmetric and positive definite for all $\boldsymbol{q}$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the coriolis and centrifugal term. $\boldsymbol{g}(\boldsymbol{q})$ is the gravity term.

If system parameters are exactly known, accurate gravity force can be computed. The gravity is not perfectly compensated through above scheme because it is difficult to know the robot parameters precisely. Hence, the estimation of the gravitational force is performed using gravity decomposition.

A. Gravity decomposition

The uncertain model parameters are estimated under quasi-static state with gravity decomposition and assumption that

[1]J.-H. Jo, D.-H. Lee, Y. Oh and S.-R. Oh is with the Center for Robotics Research, Korea Institute of Science and Technology (KIST), Seoul, 136-791, Korea (email: jhjo@kist.re.kr, phenom8305@gmail.com, oyh@kist.re.kr and sroh@kist.re.kr)

[2]Department of HCI & Robotics, UST, Daejeon, 305-350, Korea

[3]Global Technology Center, Samsung Electronics, Suwon, 443-742, Korea (email: jamestran811@gmail.com)

978-1-4799-9993-4/15/$31.00 ©2015 IEEE

the CoM of link i is located along the axial line at some distance. The gravity force $\boldsymbol{g}(\boldsymbol{q})$ can be decomposed into:

$$\boldsymbol{g}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)=\boldsymbol{Z}(\boldsymbol{q})\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$$

where $\boldsymbol{Z}(\boldsymbol{q})\in\Re^{n\times 3n(n+1)}$ is the gravity regressor which includes joint angles and $\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)\in\Re^{3n(n+1)}$ is the robot dynamic parameters. $\boldsymbol{m}$ is the vector of link masses. In addition, $\boldsymbol{p}$ and $\boldsymbol{p}_c$ are the position vector to $(i+1)$-th frame and CoM position vector of link i with respect to the i-th frame respectively. Then, $\hat{\boldsymbol{g}}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$, the estimator of $\boldsymbol{g}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$, can be obtained by

$$\begin{aligned}\hat{g}(\boldsymbol{q})&=\boldsymbol{Z}\hat{\boldsymbol{s}}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)(t)\\ \hat{\boldsymbol{s}}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)(t)&=\hat{\boldsymbol{s}}_0(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)\\ &+\int_0^t \boldsymbol{Q}\boldsymbol{Z}^{\dagger}(\boldsymbol{q})(\boldsymbol{K}_p(\boldsymbol{q}_d(\tau)-\boldsymbol{q}(\tau))-\boldsymbol{K}_d\dot{\boldsymbol{q}})d\tau\end{aligned}$$

where $\boldsymbol{Q}\in\Re^{n\times n}$ denotes an positive definite constant matrix. $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are proportional and damping gain for the gravity estimation. $\boldsymbol{q}_d(\tau)$ is the desired joint angle and $\hat{\boldsymbol{s}}_0(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$ is initially estimated value and is assumed to be zero without loss of generosity. $\boldsymbol{Z}^{\dagger}$ is the pseudo inverse of gravity regressor. The decomposition between $\boldsymbol{Z}(\boldsymbol{q})$ and $\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$ is started from computing the CoM position vector of link i.

$$\begin{aligned}\boldsymbol{p}_i^{com}&=\boldsymbol{p}_i+{}^0\boldsymbol{R}_i{}^i\boldsymbol{p}_i^{com}\\ &=\textstyle\sum_{j=0}^{i-1}{}^0\boldsymbol{R}_j{}^j\boldsymbol{p}_{j+1}+{}^0\boldsymbol{R}_i{}^i\boldsymbol{p}_i^{com}\end{aligned}$$

where $\boldsymbol{p}_i^{com}$ is the CoM position of the link i from origin and $\boldsymbol{p}_i$ is the i-th link position vector with respect to the origin. ${}^0\boldsymbol{R}_i$ is the rotation matrix from origin to i-th link and ${}^i\boldsymbol{p}_i^{com}$ is the i-th CoM vector with respect to the link i coordinate frame. ${}^i\boldsymbol{p}_{i+1}(\alpha_i,a_i,l_i)$ is the vector composed of the set of kinematic parameters from DH convention. It is expressed by

$$\boldsymbol{p}_i^{com}=\boldsymbol{R}\left(\boldsymbol{D}_i\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)$$

where

$$\boldsymbol{R}=[\boldsymbol{I}_3\quad {}^0\boldsymbol{R}_1\quad \ldots\quad {}^0\boldsymbol{R}_n]\in\Re^{3\times 3(n+1)}$$

$$\boldsymbol{p}=[{}^0\boldsymbol{p}_1^T\quad \ldots\quad {}^{n-1}\boldsymbol{p}_n^T\quad \boldsymbol{0}_3^T]^T\in\Re^{3(n+1)}$$

$$\boldsymbol{p}_c=[\boldsymbol{0}_3^T\ \ ({}^1\boldsymbol{p}_1^{com})^T\ \ \ldots\ \ ({}^n\boldsymbol{p}_n^{com})^T]^T\in\Re^{3(n+1)}$$

$$\boldsymbol{D}_i=\begin{bmatrix}\boldsymbol{I}_3^1 & & \boldsymbol{0}_3\\ & \ddots & \\ & & \boldsymbol{I}_3^i & \\ \boldsymbol{0}_3 & & & \boldsymbol{0}_3\end{bmatrix}\in\Re^{3(n+1)\times 3(n+1)}$$

$$\boldsymbol{\delta}_i=\begin{bmatrix}\boldsymbol{0}_3 & & \boldsymbol{0}\\ & \boldsymbol{I}_3^{(i,i)} & \\ \boldsymbol{0} & & \boldsymbol{0}\end{bmatrix}\in\Re^{3(n+1)\times 3(n+1)}$$

Hence, The CoM of the robot will be

$$\begin{aligned}\boldsymbol{p}_{com}&=\frac{1}{M}\sum_{i=1}^n m_i\boldsymbol{p}_i^{com}\\ &=\frac{1}{M}\sum_{i=1}^n m_i\left(\boldsymbol{p}_i+{}^0\boldsymbol{R}_i{}^i\boldsymbol{p}_i^{com}\right)\\ &=\frac{1}{M}\sum_{i=1}^n m_i\boldsymbol{R}\left(\boldsymbol{D}_i\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right).\end{aligned}$$

where M is the total mass of the robot and m_i is the mass of link i. The form of CoM jacobian matrix of link i is

$$\begin{aligned}\boldsymbol{J}_i^{com}=[&\boldsymbol{z}_1\times(\boldsymbol{p}_i^{com}-\boldsymbol{p}_1)\quad \boldsymbol{z}_2\times(\boldsymbol{p}_i^{com}-\boldsymbol{p}_2)\\ &\ldots\quad \boldsymbol{z}_i\times(\boldsymbol{p}_i^{com}-\boldsymbol{p}_i)\quad \boldsymbol{0}_{3n(n-i)}]\end{aligned}$$

where $\boldsymbol{p}_i^{com}-\boldsymbol{p}_k=\boldsymbol{R}\left[\left(\boldsymbol{D}_i-\boldsymbol{D}_k\right)\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right]=\boldsymbol{R}\left(\boldsymbol{D}_{ik}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)$. In addition,

$$\begin{aligned}\boldsymbol{J}_{com}&=\frac{1}{M}\sum_{i=1}^n m_i\boldsymbol{J}_i^{com}\\ &=\frac{1}{M}\Bigg[\sum_{i=1}^n m_i\left[\boldsymbol{z_1}\right]_\times\left(\boldsymbol{D}_{i1}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ &\qquad\sum_{i=2}^n m_i\left[\boldsymbol{z_2}\right]_\times\left(\boldsymbol{D}_{i2}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ &\qquad\ldots\quad m_n\left[\boldsymbol{z_n}\right]_\times\left(\boldsymbol{D}_{nn}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\Bigg]\end{aligned}\tag{1}$$

where $\boldsymbol{z}_1\times=[\cdot]_\times$ is the skew-symmetric operator. Using (1), gravity force can be computed by

$$\begin{aligned}\boldsymbol{g}&=Mg_c\boldsymbol{J}_{com}^T\boldsymbol{e}_z=Mg_c\boldsymbol{J}_{com}^T[0\quad 0\quad 1]^T\\ &=\begin{bmatrix}\sum_{i=1}^n m_ig_c\left(\boldsymbol{D}_{i1}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)^T\boldsymbol{R}^T\left[\boldsymbol{z_1}\right]_\times\boldsymbol{e}_z\\ \sum_{i=2}^n m_ig_c\left(\boldsymbol{D}_{i2}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)^T\boldsymbol{R}^T\left[\boldsymbol{z_2}\right]_\times\boldsymbol{e}_z\\ \ldots\\ m_ng_c\left(\boldsymbol{D}_{nn}\boldsymbol{p}+\boldsymbol{\delta}_{n+1}\boldsymbol{p}_c\right)^T\boldsymbol{R}^T\left[\boldsymbol{z_n}\right]_\times\boldsymbol{e}_z\end{bmatrix}\\ &=\begin{bmatrix}\sum_{i=1}^n\boldsymbol{e}_z^T\left[\boldsymbol{z_1}\right]_\times\boldsymbol{R}\cdot m_ig_c\left(\boldsymbol{D}_{i1}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ \sum_{i=2}^n\boldsymbol{e}_z^T\left[\boldsymbol{z_2}\right]_\times\boldsymbol{R}\cdot m_ig_c\left(\boldsymbol{D}_{i2}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ \ldots\\ \boldsymbol{e}_z^T\left[\boldsymbol{z_n}\right]_\times\boldsymbol{R}\cdot m_ng_c\left(\boldsymbol{D}_{nn}\boldsymbol{p}+\boldsymbol{\delta}_{n+1}\boldsymbol{p}_c\right)\end{bmatrix}\\ &=\boldsymbol{Z}(\boldsymbol{q})\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)\end{aligned}$$

where g_c is the gravity constant.

Therefore, the gravity regressor and dynamic parameter are chosen as

$$\boldsymbol{Z}(\boldsymbol{q})=g_c\begin{bmatrix}\boldsymbol{e}_z^T\left[\boldsymbol{z_1}\right]_\times\boldsymbol{R} & & & \boldsymbol{0}\\ & \boldsymbol{e}_z^T\left[\boldsymbol{z_2}\right]_\times\boldsymbol{R} & & \\ & & \ddots & \\ \boldsymbol{0} & & & \boldsymbol{e}_z^T\left[\boldsymbol{z_n}\right]_\times\boldsymbol{R}\end{bmatrix},$$

$$\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)=\begin{bmatrix}\sum_{i=1}^n m_i\left(\boldsymbol{D}_{i1}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ \sum_{i=2}^n m_i\left(\boldsymbol{D}_{i2}\boldsymbol{p}+\boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right)\\ \ldots\\ m_n\left(\boldsymbol{D}_{nn}\boldsymbol{p}+\boldsymbol{\delta}_{n+1}\boldsymbol{p}_c\right)\end{bmatrix}$$

In addition, generalization is extended to the multi arms in the following section.

B. Gravity decomposition for multi-arms

For body links, CoM position is computed as:

$$\boldsymbol{p}_i^{com} = \boldsymbol{p}_i + {}^0\boldsymbol{R}_i {}^i\boldsymbol{p}_i^{com} = \sum_{j=0}^{i-1} {}^0\boldsymbol{R}_j {}^j\boldsymbol{p}_{j+1} + {}^0\boldsymbol{R}_i {}^i\boldsymbol{p}_i^{com}$$

where $i = 1, \ldots, n_b$. n_b is the number of links/joints of the body. In addition to this, for links of the arm m, CoM position is computed as:

$$\begin{aligned}\boldsymbol{p}_{i+k_m}^{com} &= \sum_{j=0}^{n_b-1} {}^0\boldsymbol{R}_j {}^j\boldsymbol{p}_{j+1} + {}^0\boldsymbol{R}_{n_b} {}^{n_b}\boldsymbol{p}_{1+k_m} + {}^0\boldsymbol{R}_{1+k_m} {}^{1+k_m}\boldsymbol{p}_{1+k_m}^{com} \\ &= \sum_{j=0}^{n_b-1} {}^0\boldsymbol{R}_j {}^j\boldsymbol{p}_{j+1} + {}^0\boldsymbol{R}_{n_b} {}^{n_b}\boldsymbol{p}_{1+k_m} \sum_{j=1}^{i-1} {}^0\boldsymbol{R}_{j+k_m} {}^{k_m+j}\boldsymbol{p}_{1+k_m+j} \\ &\quad + {}^0\boldsymbol{R}_{j+k_m} {}^{j+k_m}\boldsymbol{p}_{j+k_m}^{com}\end{aligned}$$

where $i = 2, \ldots, n_a$ and n_a is the number of links/joints of an arm and

$$k_m = \begin{cases} 0 & \text{if } m = 0 \\ n_b + (m-1)\, n_a & \text{if } m > 0 \end{cases}.$$

Hence, for link i of part m, CoM position is computed as:

$$\boldsymbol{p}_{i+k_m}^{com} = \boldsymbol{R}\left(\boldsymbol{\Psi}_{i+k_m}\boldsymbol{p} + \boldsymbol{\delta}_{i+k_m+1}\boldsymbol{p}_c\right)$$

where

$$\boldsymbol{\Psi}_{i+k_m} = \begin{cases} \boldsymbol{D}_i \quad \text{if } m = 0 \ \& \ i = 1, \ldots, n_b \\ \boldsymbol{D}_{n_b} + \delta_{(n_b+1)(1+k_m)} + \boldsymbol{D}_{i+k_m} - \boldsymbol{D}_{1+k_m} \\ \quad \text{if } m > 0 \ \& \ i = 1, \ldots, n_a \end{cases}.$$

Then, CoM jacobian matrix for link i of the part m is computed as:

$$\boldsymbol{J}_{i+k_m}^{com} = \left[\boldsymbol{J}_{i+k_m}^j\right]$$

$$\boldsymbol{J}_{i+k_m}^j = \begin{cases} [\boldsymbol{z}_j]_\times \boldsymbol{R}\left(\left(\boldsymbol{\Psi}_{i+k_m} - \boldsymbol{\Psi}_j\right)\boldsymbol{p} + \boldsymbol{\delta}_{i+k_m+1}\boldsymbol{p}_c\right), & \text{conditions} \\ \boldsymbol{0}, & \text{elsewhere} \end{cases}$$

$$\text{where conditions} = \left(j \in [1, n_b] \cup [1 + k_m, k_{m+1}]\right) \cap (m > 0)$$
$$\text{or} \quad \left(j \in [1, n_b]\right) \cup (m = 0)$$

Robot CoM jacobian and gravity term can be computed with same procedure. Finally, general form of the gravity term decomposition is

$$\boldsymbol{Z} = g_c \begin{bmatrix} \boldsymbol{e}_g^T [\boldsymbol{z}_1]_\times \boldsymbol{R} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{e}_g^T [\boldsymbol{z}_2]_\times \boldsymbol{R} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & 0 & \boldsymbol{e}_g^T [\boldsymbol{z}_n]_\times \boldsymbol{R} \end{bmatrix},$$

$$\boldsymbol{s} = \begin{bmatrix}\boldsymbol{s}_0^T & \boldsymbol{s}_1^T & \ldots & \boldsymbol{s}_n^T\end{bmatrix}^T$$

where

$$\boldsymbol{s}_0 = \begin{bmatrix} \sum_{i=1}^N m_i\left(\boldsymbol{\Psi}_{i1}\boldsymbol{p} + \boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right) \\ \sum_{i=2}^N m_i\left(\boldsymbol{\Psi}_{i2}\boldsymbol{p} + \boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right) \\ \cdots \\ \\ \sum_{i=n_b}^N m_i\left(\boldsymbol{\Psi}_{in_b}\boldsymbol{p} + \boldsymbol{\delta}_{i+1}\boldsymbol{p}_c\right) \end{bmatrix},$$

$$\boldsymbol{s}_m = \begin{bmatrix} \sum_{i=1}^{n_a} m_{i+k_m}\left(\boldsymbol{\Psi}_{(i+k_m)(1+k_m)}\boldsymbol{p} + \boldsymbol{\delta}_{i+k_m+1}\boldsymbol{p}_c\right) \\ \sum_{i=2}^{n_a} m_{i+k_m}\left(\boldsymbol{\Psi}_{(i+k_m)(2+k_m)}\boldsymbol{p} + \boldsymbol{\delta}_{i+k_m+1}\boldsymbol{p}_c\right) \\ \cdots \\ m_{k_{m+1}}\left(\boldsymbol{\Psi}_{k_{m+1}k_{m+1}}\boldsymbol{p} + \boldsymbol{\delta}_{k_{m+1}+1}\boldsymbol{p}_c\right) \end{bmatrix}$$

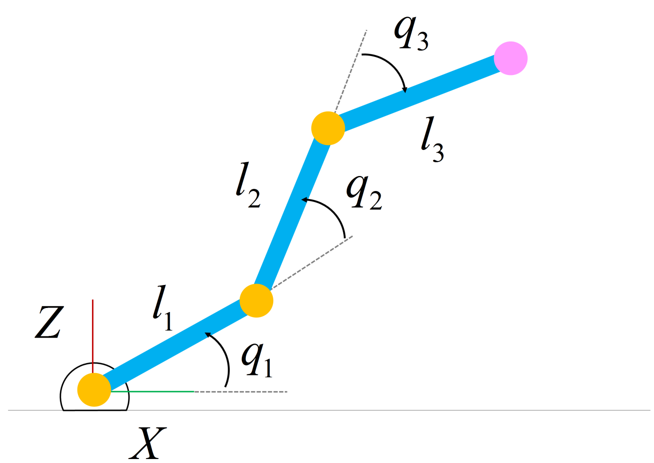


Fig. 1. 3 degrees of freedom planar manipulator.

C. 3D planar manipulator

This is a simple example of the gravity decomposition. Let it be 3 DoFs planar manipulator as shown in Fig. 1 assuming that the CoM of link i is located along the axial line at some distance l_{gi}.

In addition, the position vector of the link is:

$$\boldsymbol{p}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{p}_2 = \begin{bmatrix} l_1 c_1 \\ 0 \\ l_1 s_1 \end{bmatrix}, \quad \boldsymbol{p}_3 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ 0 \\ l_1 s_1 + l_2 s_{12} \end{bmatrix},$$

$$\boldsymbol{p}_4 = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ 0 \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix}, \boldsymbol{z}_1 = \boldsymbol{z}_2 = \boldsymbol{z}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}.$$

where $c_1 = cos(q_1)$, $c_{12} = cos(q_1 + q_2)$, $c_{123} = cos(q_1 + q_2 + q_3)$, $s_1 = sin(q_1)$, $s_{12} = sin(q_1 + q_2)$, $s_{123} = sin(q_1 + q_2 + q_3)$. With those parameters, $\boldsymbol{Z}(\boldsymbol{q}), \boldsymbol{s}(\boldsymbol{m}, \boldsymbol{p}, \boldsymbol{p}_c)$ can be computed as shown in Fig. 2.

$$\boldsymbol{Z}(\boldsymbol{q}) = g\left[\boldsymbol{Z}_1 \ \boldsymbol{Z}_2 \ \boldsymbol{Z}_3\right]$$
$$\boldsymbol{s}(\boldsymbol{m}, \boldsymbol{p}, \boldsymbol{p}_c) = [\boldsymbol{0}_3 \ \ \boldsymbol{s}_1^T \ \ \boldsymbol{0}_2 \ \ \boldsymbol{s}_2^T \ \ \boldsymbol{0}_2 \ \ \boldsymbol{s}_3^T$$
$$\ldots \ \ \boldsymbol{0}_5 \ \ \boldsymbol{s}_2^T \ \ \boldsymbol{0}_2 \ \ \boldsymbol{s}_3^T \ \ \boldsymbol{0}_8 \ \ \boldsymbol{s}_3^T \ \ \boldsymbol{0}_2]^T$$

Delete zeros

$$\boldsymbol{s} = [\boldsymbol{0}_3 \;\; l_1(m_2+m_3)+l_{g1}m1 \;\; \boldsymbol{0}_2 \;\; l_2m_3+l_{g2}m_2 \;\; \boldsymbol{0}_2 \;\; l_{g3}m_3 \;\; \boldsymbol{0}_8 \;\; l_2m_3+l_{g2}m_2 \;\; \boldsymbol{0}_2 \;\; l_{g3}m_3 \;\; \boldsymbol{0}_{11} \;\; l_{g3}m_3 \;\; \boldsymbol{0}_2]$$

Merge same elements

(a) Deleting zeroes and merging same elements of $\boldsymbol{s}$ for compact form.

$$\boldsymbol{Z} = g_c \begin{bmatrix} 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 \end{bmatrix}$$

+

+

(b) Deleting corresponding columns to the deleted zeroes and merging colmuns of $\boldsymbol{Z}$ for compact form.

Fig. 2. Compact form derivation of gravity force of 3 DoFs manipulator.

where
$$\boldsymbol{Z}_1 = \begin{bmatrix} 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix},$$
$$\boldsymbol{Z}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\boldsymbol{Z}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & c_1 & -s_1 & 0 & c_{12} & -s_{12} & 0 & c_{123} & -s_{123} & 0 \end{bmatrix},$$

$$\begin{aligned} \boldsymbol{s}_1 &= l_1(m_2+m_3)+l_{g1}m_1, \\ \boldsymbol{s}_2 &= l_2m_3+l_{g2}m_2, \\ \boldsymbol{s}_3 &= l_{g3}m_3. \end{aligned}$$

These repeated terms can be simplified through the deleting and merging of the columns in the derived equation. As shown in the Fig. 2, the zero columns in $\boldsymbol{s}$ and corresponding rows in $\boldsymbol{Z}$ are deleted for reduction of the computation and same elements in $\boldsymbol{Z}$ are merged into one column. Through the computation, compact form can be obtained and makes computation time reduced. Hence, the compact form of the gravity decomposition becomes

$$\boldsymbol{g}(\boldsymbol{q},\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c) = \boldsymbol{Z}(\boldsymbol{q})\boldsymbol{s}(\boldsymbol{m},\boldsymbol{p},\boldsymbol{p}_c)$$

$$= g \begin{bmatrix} c_1 & c_{12} & c_{123} \\ 0 & c_{12} & c_{123} \\ 0 & 0 & c_{123} \end{bmatrix} \begin{bmatrix} l_1\left(m2+m_3\right)+l_{g_1}m_1 \\ l_2m_3+l_{g_2}m_2 \\ l_{g_3}m_3 \end{bmatrix}$$

$$= g \begin{bmatrix} c_1\left(l_1(m_2+m_3)+l_{g1}m_1\right)+c_{12}\left(l_2m_3+l_{g2}m_2\right)+c_{123}l_{g3}m_3 \\ c_{12}\left(l_2m_3+l_{g2}m_2\right)+c_{123}l_{g3}m_3 \\ c_{123}l_{g3}m_3 \end{bmatrix}.$$

III. Experimental results

The algorithm can be implemented in any kind of robot according to II-B. It can be used instead of approximated gravity obtained from partial derivative of potential energy. Hence, estimation can be performed in the initial state before manipulation and once the gravity force is estimated through the decomposition in the initial state. The important point in

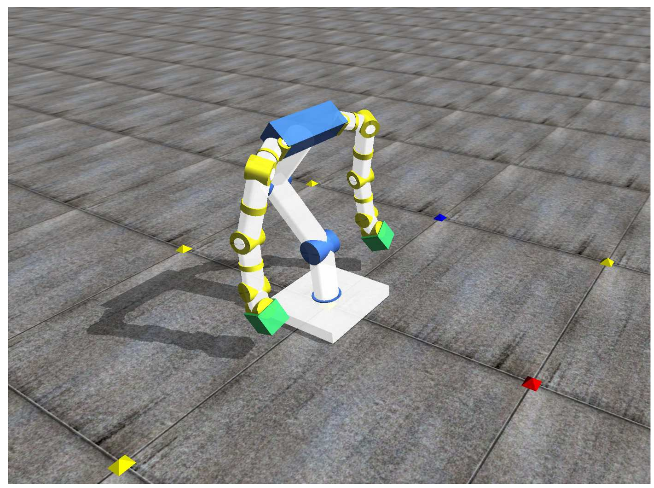


Fig. 3. 19 degrees of freedom upper-body dual arm robot.

the estimation state is that the robot joints should endure links falling force by gravity. If the links configurationally do not endure the gravity force, the estimation cannot be performed appropriately. Hence, the estimation should be performed under quasi-static state with gravity-endurable configuration before the robot manipulation.

The following section shows the several experiments and its results for the comparison of the performance.

A. Upper-body dual arm

The proposed algorithm is implemented on the upper-body dual arm robot which has 19 DoFs of rotary actuator including 8 for each arm and 3 for the trunk as shown in Fig. 3. The weight and length of the robot are approximately 62kg, 100cm and 65cm. The sampling rate is 1kHz and Open Dynamic Engine(ODE) is used for the simulation. The robot performed joint PD control with with and without gravity estimation for the comparison of performance of the proposed control. In all simulations, the task is reaching in the free space from origin to desired position including estimation pos approaching(0.2-7s), estimation(7-12s), homing(12-17s), approaching 1(17-21s), approaching 2(22-26s), and homing(27-31s). Since this study is not focusing on trajectory planning, so this paper compares the position error only when the algorithm is implemented or not.

B. PD control with and without gravity estimation

First, joint PD controller without any compensator is used to perform the reaching task. The control torque is

$$\boldsymbol{\tau} = \boldsymbol{K}_p\left(\boldsymbol{q}_d - \boldsymbol{q}\right) - \boldsymbol{K}_d\dot{\boldsymbol{q}}$$

Each figure in Fig. 4 shows the position error of the trunk in Fig. 4. (a), left arm in Fig. 4. (b), and right arm in Fig. 4. (c) when the gravity estimation is not conducted. Then, joint PD controller with estimated gravity is used to perform the reaching task after the gravity estimation. The controller of the second experiment is

$$\boldsymbol{\tau} = \boldsymbol{K}_p\left(\boldsymbol{q}_d - \boldsymbol{q}\right) - \boldsymbol{K}_d\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}).$$

Fig. 5 shows the errors of trunk, left arm, and right arm when the gravity estimation algorithm is implemented on the robot. In the estimation state, as the gravity estimated, the position error reduces in 7-12s. In the following, when the robot approaches to the desired position, it makes some error however it can be reduced by estimating the gravity force for long time more than this trial. In the experiment, gains of the robot are quite huge not as same as usual in the practical robot. Because non-linear effects such as friction are not considered so that the result of first experiment looks quite nice. However, this control scheme will show worse performance in real robot system due to the uncertainties.

IV. DISCUSSION AND CONCLUSION

This paper shows that gravity estimation through the decomposition by generalized gravity regressor and the uncertain model parameters and its application to the robot control. The motivation of the paper is that there is the difficulties to set the equation of motion due to the uncertain dynamic parameters. Hence, the estimation of the gravity is condcted and main goal is the generalization of the gravity decomposition forming by the gravity regressor and dynamic parameters regardlessly to the any kind of robot system. The inverse of gravity regressor is used to compute the uncertain dynamic parameters. Additionally, the gravity decomposition of the upper-body dual arm robot whose 19 DoFs and the compact form either shown as an example. Another advantage of this method is that the generalized form is once decomposed, the process is no longer necessary to be derived.

As a future work, the algorithm will be implemented in the real robot, and the generalized form can be extended to estimating the real CoM location of a robot and detecting additional weight not only at end-effector but at anywhere on the robot.

REFERENCES

[1] H. Cheng, Y.-K. Yiu, and Z. Li. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on Mechatronics*, 8(4):483–491, Dec 2003.

[2] L. Sciavicco and B. Siciliano. *Modelling and control of robot manipulators*. Springer Verlag, 2000.

[3] B. Bona and M. Indri. Friction compensation in robotics: an overview. In *44th IEEE Conference on Decision and Control and 2005 European Control Conference(CDC-ECC'05)*, pages 4360–4367, Dec 2005.

[4] S. Arimoto, H. Hashiguchi, and M. Sekimoto. Natural resolution of dof redundancy in execution of robot tasks under the gravity: a challenge to bernstein's problem and applications to handwriting robots. In *in Proc. 12th International Conference on Advanced Robotics(ICAR'05)*, pages 51–57, July 2005.

[5] M.W. Spong. On the robust control of robot manipulators. *IEEE Transactions on Automatic Control*, 37(11):1782–1786, Nov 1992.

[6] S. ARIMOTO. Optimal linear quadratic regulators for control of nonlinear mechanical systems with redundant degrees-of-freedom. *SICE Journal of Control, Measurement, and System Integration*, 4(4):289–294, 2011.

[7] C.C. Cheah and H.C. Liaw. Inverse jacobian regulator with gravity compensation: Stability and experiment. *IEEE Transactions on Robotics*, 21(4):741–747, Aug 2005.

[8] C.C. Cheah, C. Liu, and H.C. Liaw. Stability of inverse jacobian control for robot manipulator. In *in Proc. 2004 IEEE International Conference on Control Applications*, volume 1, pages 321–326 Vol.1, Sept 2004.

[9] H. Yazarel, C.C. Cheah, and H.C. Liaw. Adaptive sp-d control of a robotic manipulator in the presence of modeling error in a gravity regressor matrix: theory and experiment. *IEEE Transactions on Robotics and Automation*, 18(3):373–379, Jun 2002.

[10] H. Yazarel and C.C. Cheah. Adaptive visual servoing of robots with uncertain gravity regressor and jacobian matrices. In *in Proc of the 2001 American Control Conference*, volume 1, pages 652–657 vol.1, 2001.

[11] C.C. Cheah, S. Kawamura, S. Arimoto, and K. Lee. Pid control of robotic manipulator with uncertain jacobian matrix. In *in Proc. 1999 IEEE International Conference on Robotics and Automation(ICRA'99)*, volume 1, pages 494–499 vol.1, 1999.

[12] C. Liu and C. C. Cheah. Task-space adaptive setpoint control for robots with uncertain kinematics and actuator model. *IEEE Transactions on Automatic Control*, 50(11):1854–1860, Nov 2005.

[13] C. Liu and C Chern Cheah. Adaptive regulation of rigid-link electrically driven robots with uncertain kinematics. In *in Proc of the 2005 IEEE International Conference on Robotics and Automation(ICRA'05)*, pages 3262–3267, April 2005.

[14] M. Galicki. An adaptive regulator of robotic manipulators in the task space. *IEEE Transactions on Automatic Control*, 53(4):1058–1062, May 2008.

[15] C.C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto. Approximate jacobian control with task-space damping for robot manipulators. *IEEE Transactions on Automatic Control*, 49(5):752–757, May 2004.

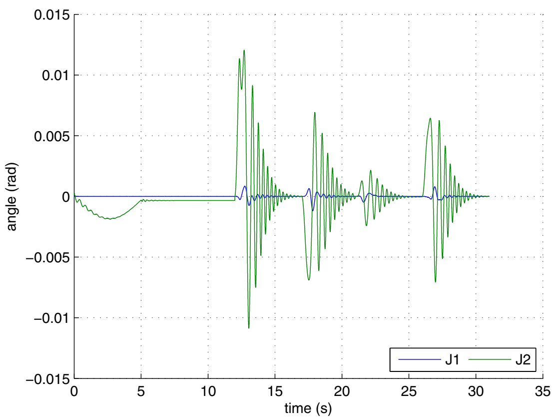
[16] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto. Approximate jacobian control for robots with uncertain kinematics and dynamics. *IEEE Transactions on Robotics and Automation*, 19(4):692–702, Aug 2003.

[17] H. Yazarel and C.C. Cheah. Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics. *IEEE Transactions on Automatic Control*, 47(9):1580–1585, Sep 2002.

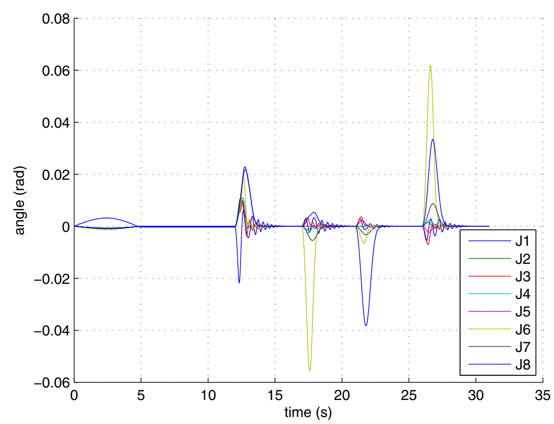
[18] C.C. Cheah, S. Kawamura, and S. Arimoto. Stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties. *Automatica*, 39(5):847 – 855, 2003.

[19] Y. Zhao and C.C. Cheah. Hybrid vision-force control for robot with uncertainties. In *in Proc. 2004 IEEE International Conference on Robotics and Automation(ICRA'04)*, volume 1, pages 261–266 Vol.1, April 2004.

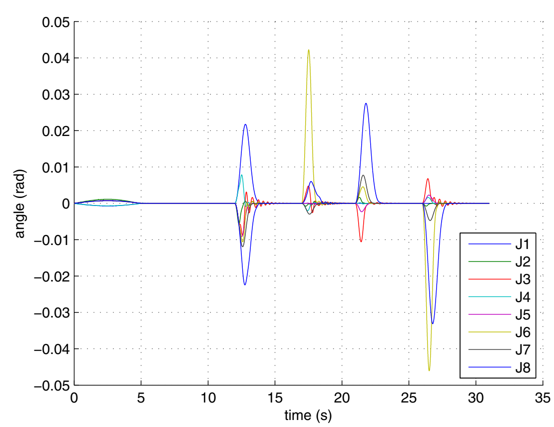
[20] C.-S. Chiu, K.-Y. Lian, and T.-C. Wu. Robust adaptive motion/force tracking control design for uncertain constrained robot manipulators. *Automatica*, 40(12):2111 – 2119, 2004.



(a) Position error of trunk joints.

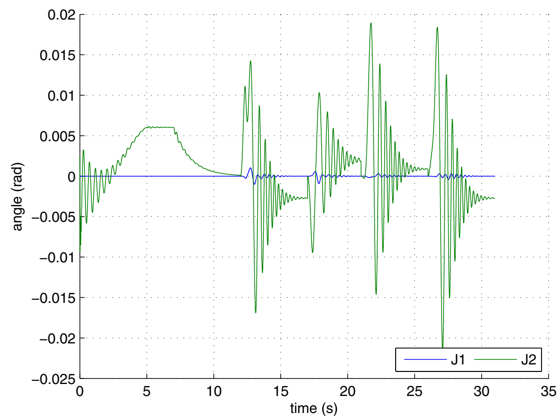


(b) Position error of left arm joints.

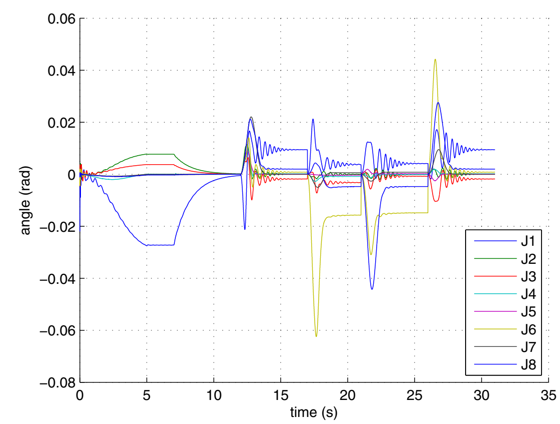


(c) Position error of right arm joints.

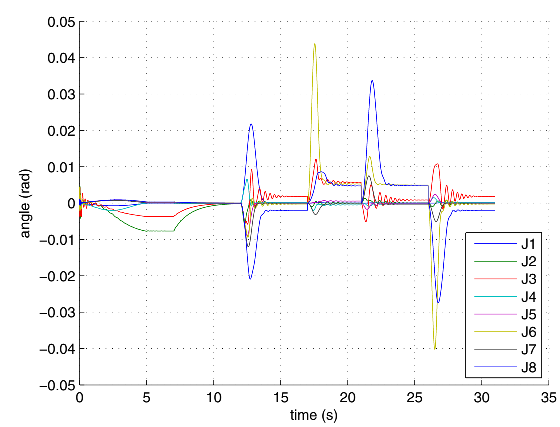
Fig. 4. Joints errors of the robot without gravity estimation.



(a) Position error of trunk joints.



(b) Position error of left arm joints.



(c) Position error of right arm joints.

Fig. 5. Joints errors of the robot with gravity estimation.