Robot Calibration

- Calibration 의 목적

  Improves robot accuracy through software rather than changing the mechanical Structure or design of Robot

- Calibration Procedure

  modeling, measurement, identification, correction

- 3 levels of Robot Calibration

  ① level 1 Calibration : Joint level Calibration

  - determines the correct relationship between Joint displacement transducer signals and the actual joint displacement

  - $\theta_i = h_i(\eta_i, r_i)$

    θ_i : actual joint displacement

    η_i : signal from the transducer

    r_i : parameters in the fn. $h_i(\ )$.

- 대부분의 경우, h( )는 linear로 model됨

$$\theta_i = K_{i1}\eta_i + K_{i2}$$

- Gear Backlash + Joint Clearance 와 같은 Nonlinear effects는 $\left\{ \begin{array}{l} \text{nonkinematic model로 level 3에서 model.} \\ \text{Kinematic model 이지만 (경우에 따라 적용)} \end{array} \right.$

② level 2 Calibration : Entire Robot Kinematic Model Calibration

- determines the basic Kinematic geometry of the robot as well as the correct joint angle relationships

- $\underline{x} = g(\eta, r, a)$

  η : joint transducer readings

  r : Coefficients of Joint level Calibration model

  a : Coefficients in the Kinematic model

③ level 3 Calibration : nonkinematic (nongeometric) Calibration

- NonKinematic Error 요소 보정

  Joint Compliance, Friction, Clearance and Link compliance

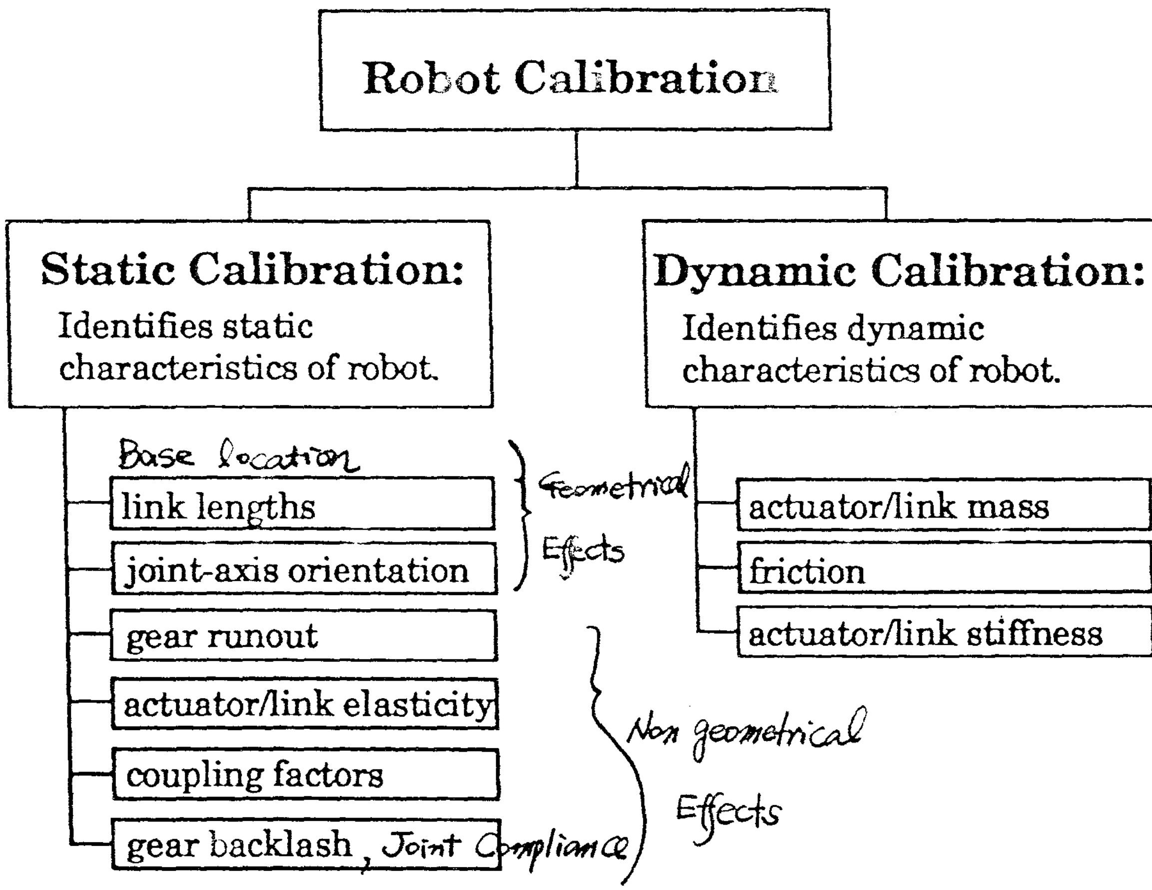
- Dynamic Model Parameters

- $x = l(\eta, r, a, \dot{x}, \ddot{x}, \mu, \beta)$

  μ : coefficients in the dynamic model

  β : coefficients in Nonkinematic error factor model

- 앞의 3 levels of Calibration을 Static Calibration과 Dynamic Calibration으로 분류하는 것이 일반적임.



- 일반적으로 Static Calibration을 Kinematic Calibration 이라고도 함

- Kinematic Calibration은 Robot Kinematic model의 정확성을 이루어 Robot Accuracy 향상.

  Ex) Puma 제작 후 Accuracies : about 1 cm

  Calibration 후 Accuracies : about 0.2 ~ 0.3 mm

- 본 연구의 주요 관심은 level 2 Calibration 즉 Kinematic Calibration 중 Geometrical Effect 만을 고려함.

- Kinematic Calibration of Manipulator

    - Assumed to be rigid link
    - Assumed to be perfect joint (No joint clearance)
    - Denavit - Hartenburg Kinematic model

$$X = f(\theta, \alpha, a, d) \quad \text{—①}$$

$\theta \to d \to \alpha \to a$ (Paul)

$\alpha \to a \to \theta \to d$ (Craig)

i) X를 알고 θ를 구하는 것 (α, a, d는 given) : Inverse Kinematic

ii) Inverse Kinematics 자체도 highly Nonlinear 이므로 해를 구하는데 어려움이 많다.

- ①식을 현재의 nominal values 에 대하여 linearization

$$\Delta X = \frac{\partial f}{\partial \theta}\Delta\theta + \frac{\partial f}{\partial \alpha}\Delta\alpha + \frac{\partial f}{\partial a}\Delta a + \frac{\partial f}{\partial d}\Delta d$$

$$= \left[\frac{\partial f}{\partial \theta} \;\vdots\; \frac{\partial f}{\partial \alpha} \;\vdots\; \frac{\partial f}{\partial a} \;\vdots\; \frac{\partial f}{\partial d} \right] \begin{bmatrix} \Delta\theta \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix}$$

$$= \left[J_\theta \;\vdots\; J_\alpha \;\vdots\; J_a \;\vdots\; J_d \right] \begin{bmatrix} \Delta\theta \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix}$$

현재의 nominal values 에 의하여 X 값을 구하면 $X_{computed}$ 측정한 X 값은 $X_{measured}$ 그리고 $J_\theta, J_\alpha, J_a, J_d$ 를 구한다.

$$\therefore \Delta X = X_m - X_c = \underbrace{\left[J_\theta \;\vdots\; J_\alpha \;\vdots\; J_a \;\vdots\; J_d \right]}_{C} \underbrace{\begin{bmatrix} \Delta\theta \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix}}_{\Delta\phi}$$

◦ 일반적으로 manipulator의 link parameter는 24개 이므로 최소한 4개 이상의 측정점 필요. 5개의 측정점에서 Position과 Orientation 모두 측정되었다면

$$\Delta X_{30\times 1} = C_{30\times 24}\,\Delta\phi_{24\times 1}$$

위 Eq.을 만족하는 $\Delta\phi$를 구하기 위하여 $e \triangleq \Delta X - C\Delta\phi$

$$\min \tfrac{1}{2} e^T e \triangleq L$$

$$L(\Delta\phi) = \frac{1}{2}(\Delta X - C\Delta\phi)^T(\Delta X - C\Delta\phi)$$

$$\frac{\partial L}{\partial \Delta\phi} = -(\Delta X - C\Delta\phi)^T C = 0$$

$$= \Delta\phi^T C^T C - \Delta X^T C = 0$$

$$C^T C\,\Delta\phi = C^T \Delta X$$

$$\therefore\ \underline{\underline{\Delta\phi = (C^T C)^{-1} C^T \Delta X}}$$

Error least square soln.

- Iterative least square algorithm

  i) $\Delta X_i = C_i \, \Delta\phi_i$

  현재의 $\theta_i, \alpha_i, d_i, a_i$ 에 의해서 X_{com} , C_i 계산

  ii) $\Delta\phi_i = (C_i^T C)^{-1} C^T \Delta X_i$ ← least square soln.

  iii) $\phi_{i+1} = \phi_i + \Delta\phi_i$

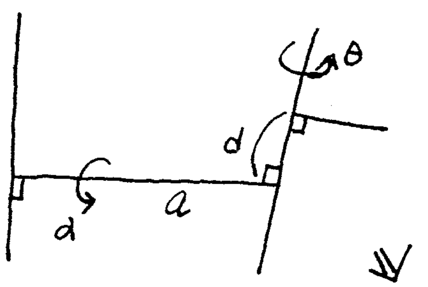
  iv) back to i)

- 제시한 알고리즘의 문제점

  ① Neighboring joint axes are nearly parallel 한 경우

  (Hayati ; Mooring)

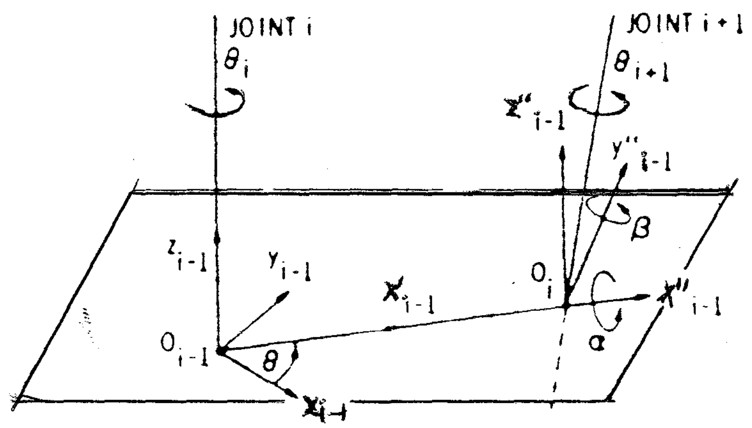
  - Common normal is extremely sensitive to slight variations in axis rotation and its location will vary wildly



d의 작은 변화에 d값의 변화가 심함

⇓

- Modified D-H parameters :



② Iteration step ii) 에서 $(C^TC)^{-1}$ 의한 Singularity Prob.

- Initial guess 값이 많이 틀리면 중간 계산 과정에서 Singularity 발생 : Transient Singularity

- Levenberg-Marquart Algorithm or Singularity Robust Algorithm 사용.

$$\min \frac{1}{2}e^Te + \frac{1}{2}\lambda \Delta\phi^T\Delta\phi$$

$$L(\Delta\phi, \lambda) = \frac{1}{2}(\Delta X - C\Delta\phi)^T(\Delta X - C\Delta\phi) + \frac{\lambda}{2}\Delta\phi^T\Delta\phi$$

$$\frac{\partial L}{\partial \Delta\phi} = -(\Delta X - C\Delta\phi)^T C + \lambda\Delta\phi^T = 0$$

$$= \Delta\phi^T C^T C + \lambda\Delta\phi^T - \Delta X^T C = 0$$

$$(C^TC + \lambda I)\Delta\phi = C^T\Delta X$$

$$\therefore \quad \underline{\underline{\Delta\phi = (C^TC + \lambda I)^{-1} C^T \Delta X}}$$

Singurarity Robust Inverse

λ : scale factor between the exactness and the fesibility of the soln.

- 일반적으로 초기 step에서 좋은 수렴 특성을 보이나 최종 step에서 slow convergence를 보임.

- A Method for automatically choosing the scale factor λ

  - Singularity 부분에서는 λ가 큰값을 가져야 하지만 Singularity 에서 멀어진 부분에서는 $\lambda = 0$ 로 하는 switching technique 고안.

  - Nakamura 의 방법

$$\lambda = \begin{cases} \lambda_0 \left(1 - \frac{\omega}{\omega_0}\right) & \text{if } \omega < \omega_0 \\ 0 & \text{otherwise} \end{cases}$$

$\omega = \sqrt{\det(C^T C)}$: manipulability measure

ω_0 : singular region boundary

λ : singular point 에서 λ의 적절한 값

  - Kelma & Khosla 의 방법

i) ω is dependent on manipulator size

ii) a_j 나 D_j 중 가장 큰값으로 나머지 DH parameter를 나누면 ω 값은 0과 1 사이가 된다.

$0 < \omega < 1$

$$\lambda = \begin{cases} \lambda_0 \left(1 - \frac{\omega_{K+1}}{\omega_K}\right) & \text{if } \frac{\omega_{K+1}}{\omega_K} < \mu \\ 0 & \end{cases}$$

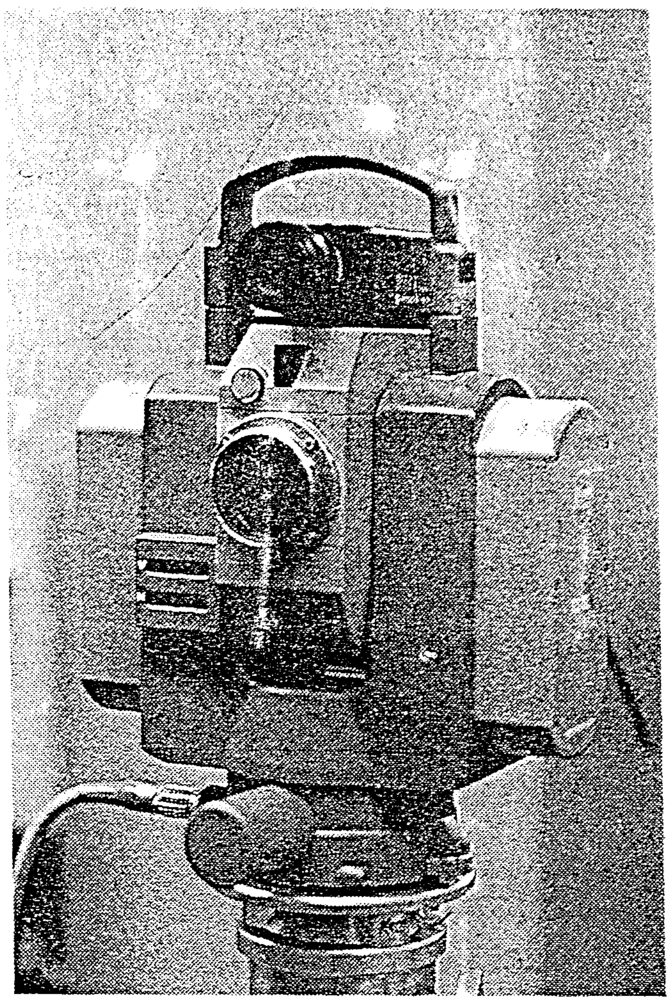
실험에 의하면 μ 값은 $\frac{1}{\sqrt{10}}$ 이 좋다.

◎ Measurement of the end-point location

- 일반적으로 3차원 측정은 어렵고, 고가의 장비 요구

- Coordinate Measuring Machine [Driel , Zhong]

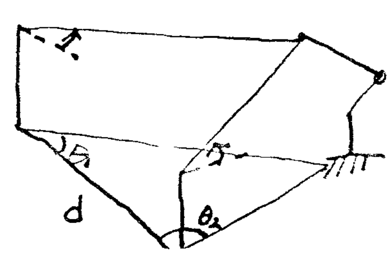
⊙ Theodolites [Whitney, Lozinski and Rourke]



CCD Vision
with telescope 형태로
주로 point detection
with high precision

- Extremely good results (0.1mm) but need the extensive human involvement (taking 5 days)

- Principle of triangulation



- Special fixture [Tang] — using 'free mode' of robot
- laser interferometers [Prenninger]

◎ Identifiability

- 몇 개의 점을 측정하여야 하는가?
- 측정 pose는 얼마나 다양하여야 하는가?
- (중요 결과)

  Complete Parameter Identification을 위해서

  partial Pose Information 만으로도 충분하다.

  [Goswami; Driel; Veitschegger]

- ~~Automotive~~ Autonomous Calibration의 가능성 제시

  special fixture 사용 가능

◎ Autonomous Calibration with special devices

- 3 dimensional motion tracking system [Hollerbach, Bennet]
- laser range finders [Lau, Hooken, Haynes]

위에서 언급한

◎ Kinematic Calibration의 결과는 보통 0.2 ~ 0.3 mm

0.01 mm ~ 0.02 mm 의 결과를 얻으려면 link flexibility 나

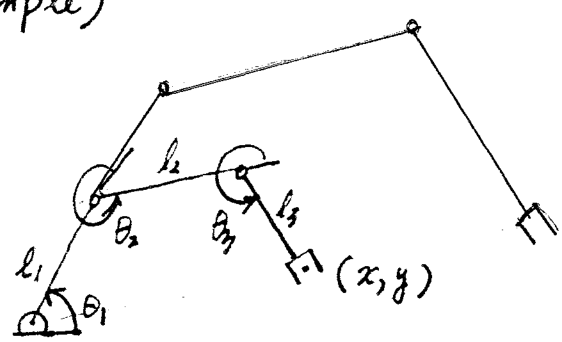
Thermal effect 고려

◎ Autonomous Calibration

- On-site Calibration이 필요 : Industry
- External Sensor를 사용하지 않고 Internal Sensor 만을 사용 : use of only joint readings
- ~~Hand~~ Autonomous Hand / Eye Calibration :

  use of only joint readings and Camera measurements

Easy Example)



$$l_1 \cos\theta_1 + l_2 \cos\theta_{12} + l_3 \cos\theta_{123} = x \qquad \theta_{12} = \theta_1 + \theta_2$$

$$l_1 \sin\theta_1 + l_2 \sin\theta_{12} + l_3 \sin\theta_{123} = y$$

고정된 x, y 에 대하여 manipulator를 움직여 l_1, l_2, l_3를 구하고자 한다. 이 경우, l_1 l_2 l_3 의 any scale 도 위 Eq.을 만족하기 때문에 이중 하나를 fix하면 (ex. $l_1 = 1$), 3번 측정

$$\begin{bmatrix} \cos\theta_{12}^1 & \cos\theta_{123}^1 & -1 & 0 \\ \sin\theta_{12}^1 & \sin\theta_{123}^1 & 0 & -1 \\ \cos\theta_{12}^2 & \cos\theta_{123}^2 & -1 & 0 \\ \sin\theta_{12}^2 & \sin\theta_{123}^2 & 0 & -1 \\ \cos\theta_{12}^3 & \cos\theta_{123}^3 & -1 & 0 \\ \sin\theta_{12}^3 & \sin\theta_{123}^3 & 0 & -1 \end{bmatrix} \begin{bmatrix} l_2 \\ l_3 \\ x \\ y \end{bmatrix} = \begin{bmatrix} -\cos\theta_1^1 \\ -\sin\theta_1^1 \\ -\cos\theta_1^2 \\ -\sin\theta_1^2 \\ -\cos\theta_1^3 \\ -\sin\theta_1^3 \end{bmatrix}$$

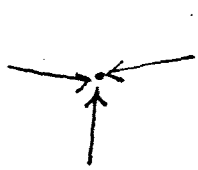
$C\phi = r$ 의 형태

$\phi = (C^TC)^{-1} C^T r$ 로서 l_1에 대한 l_2, l_3, x, y를 구할 수 있다.

• 이 경우 평면 문제 이므로 Nonlinear Eq.을 직접 사용 3차원에서는 불가능

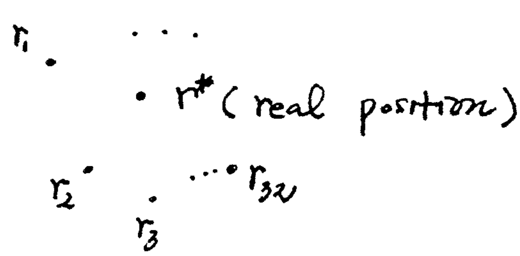
⊙ 현대 Robot에서 사용하고 있는 자동 Tool 정수 보정 방법의 추정

: Fixed Point에 근거한 Calibration Algorithm



① Work Space 한점에 대하여 32회 teaching 그 결과 $\theta_1 \sim \theta_{32}$ 까지의 Joint Reading Set을 얻음.

② 현재의 norminal value에 의하여 End-Effector Position 구함 ($\underline{r}_1 \sim \underline{r}_{32}$)



③ $$r_1 - r^* = \Delta r_1 = [J_{\phi_1}]\Delta\phi = [J_{\theta_1} \vdots J_\alpha \vdots J_a \vdots J_d]\begin{bmatrix}\Delta\theta_{offset}\\ \Delta\alpha \\ \Delta a \\ \Delta d\end{bmatrix}$$

$$\phi = \begin{bmatrix}\theta\\ \alpha \\ a \\ d\end{bmatrix}$$

$\theta = \theta_R + \underline{\theta_{offset}}$ 이라고 가정

→ constant nongeometric effect

④ ③과 같은 방법으로

$$r_2 - r^* = [J_{\phi_2}]\,\Delta\phi$$

$$r_3 - r^* = [J_{\phi_3}]\,\Delta\phi$$

$$\vdots$$

$$r_{32} - r^* = [J_{\phi_{32}}]\,\Delta\phi$$

⑤ $$(r_1 - r^*) - (r_2 - r^*) = [J_{\phi_1}]\Delta\phi - [J_{\phi_2}]\Delta\phi$$

$$r_1 - r_2 = [J_{\phi_1} - J_{\phi_2}]\,\Delta\phi$$

같은 방법으로,

$$r_1 - r_3 = [J_{\phi_1} - J_{\phi_3}]\,\Delta\phi$$

$$\vdots$$

$$r_1 - r_{32} = [J_{\phi_1} - J_{\phi_{32}}]\,\Delta\phi$$

종합하면,

$$\begin{bmatrix} r_1 - r_2 \\ r_1 - r_3 \\ \vdots \\ r_1 - r_{32} \end{bmatrix} = \begin{bmatrix} J_{\phi_1} - J_{\phi_2} \\ J_{\phi_1} - J_{\phi_3} \\ \vdots \\ J_{\phi_1} - J_{\phi_{32}} \end{bmatrix} \Delta\phi$$

$$\therefore$$

$$\Delta X = C\,\Delta\phi \quad \rightarrow \quad \Delta\phi = (C^T C)^{-1} C^T \Delta X$$

Iterative least square Algorithm 적용

A New Method for Autonomous Robot Calibration

Zhong and Lewis

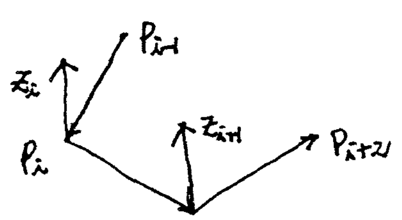
- Partial Pose information is sufficient for complete parameter identification
- Tang : a special fixture to obtain partial pose information of a robot end-effector : point and orientation angle measure
  using ~~robot~~ the 'free mode' of robot

그러나 모든 robot은 free mode를 제공하지 않는다.

- on-site calibration in an industrial environment

2. Formulation of the Kinematic Identification Model

$$P_i = P_i^0 + J_i \Delta\rho = [x_i^0 \; y_i^0 \; z_i^0]^T + [J_i^x \; J_i^y \; J_i^z]^T \Delta\rho \qquad (1)$$



coplanar points

$$\Delta P_i = P_i - P_{i-1} = [\Delta x_i^0 \;\; \Delta y_i^0 \;\; \Delta z_i^0] + [\Delta J_i^x \;\; \Delta J_i^y \;\; \Delta J_i^z]^T \Delta\rho$$

$$\Delta x_i^0 = x_i^0 - x_{i-1}^0 \; ; \quad \Delta y_i^0 = y_i^0 - y_{i-1}^0 \; ; \quad \Delta z_i^0 = z_i^0 - z_{i-1}^0$$

$$\Delta J_i^x = J_i^x - J_{i-1}^x \; ; \quad \Delta J_i^y = J_i^y - J_{i-1}^y \; ; \quad \Delta J_i^z = J_i^z - J_{i-1}^z$$

$$Z_i = \Delta P_{i+1} \times \Delta P_i = [a0_i\ b0_i\ c0_i]^T + [a1_i\ b1_i\ c1_i]^T \Delta\rho \quad (3)$$

where

$a0_i = \Delta y_i^0 \Delta z_{i+1}^0 - \Delta y_{i+1}^0 \Delta z_i^0;$
$b0_i = \Delta z_i^0 \Delta x_{i+1}^0 - \Delta z_{i+1}^0 \Delta x_i^0;$
$c0_i = \Delta x_i^0 \Delta y_{i+1}^0 - \Delta x_{i+1}^0 \Delta y_i^0;$
$a1_i = (\Delta y_i^0 \Delta J_{i+1}^z + \Delta z_{i+1}^0 \Delta J_i^y) - (\Delta y_{i+1}^0 \Delta J_i^z + \Delta z_i^0 \Delta J_{i+1}^y);$
$b1_i = (\Delta z_i^0 \Delta J_{i+1}^x + \Delta x_{i+1}^0 \Delta J_i^z) - (\Delta z_{i+1}^0 \Delta J_i^x + \Delta x_i^0 \Delta J_{i+1}^z);$
$c1_i = (\Delta x_i^0 \Delta J_{i+1}^y + \Delta y_{i+1}^0 \Delta J_i^x) - (\Delta x_{i+1}^0 \Delta J_i^y + \Delta y_i^0 \Delta J_{i+1}^x).$

참고

$$\left\{ \begin{aligned} &(a + A\rho) \times (b + A'\rho) \\ &= a \times b + a \times A'\rho + A\rho \times b + A\rho \times A\rho' \end{aligned} \right.$$

$$Z_{i+1} = [a0_{i+1}\ b0_{i+1}\ c0_{i+1}]^T + [a1_{i+1}\ b1_{i+1}\ c1_{i+1}]^T \Delta\rho. \quad (4)$$

$$\begin{aligned} Z_{i+1} \times Z_i &= [A0_i\ B0_i\ C0_i]^T + [A1_i\ B1_i\ C1_i]^T \Delta\rho \\ &= 0 \end{aligned} \quad (5)$$

where

$A0_i = b0_i c0_{i+1} - b0_{i+1} c0_i;$
$B0_i = c0_i a0_{i+1} - c0_{i+1} a0_i;$
$C0_i = a0_i b0_{i+1} - a0_{i+1} b0_i;$
$A1_i = (b0_i c1_{i+1} + c0_{i+1} b1_i) - (b0_{i+1} c1_i + c0_i b1_{i+1});$
$B1_i = (c0_i^0 a1_{i+1} + a0_{i+1} c1_i) - (c0_{i+1} a1_i + a0_i c1_{i+1});$
$C1_i = (a0_i b1_{i+1} + b0_{i+1} a1_i) - (a0_{i+1} b1_i + b0_i a1_{i+1}).$

Denoting $H = [A1_1\ B1_1\ C1_1 \ldots A1_m\ B1_m\ C1_m]^T$; and
$\Delta X = [A0_1\ B0_1\ C0_1 \ldots A0_m\ B0_m\ C0_m]^T$,

We have a linear system:

$$H\Delta\rho + \Delta X = 0 \quad (6)$$

- Robot의 Base x-y plane과 Constraint plane을 일치시키면

$$P_i = P_i^\circ + J_i \Delta\rho = [x_i^\circ\ y_i^\circ\ z_i^\circ]^T + [J_i^x\ J_i^y\ J_i^z]\Delta\rho$$

여기서, $z_i^\circ + J_i^z \Delta\rho = 0$ 간단한 Eq. 도출

게다가, $Z_i = \Delta P_{i+1} \times \Delta P_i$ 의 x, y Component 0

- Constraint plane에 대한 정확한 정보가 없다면

$$H\Delta\rho + \Delta X = 0$$ 을 사용

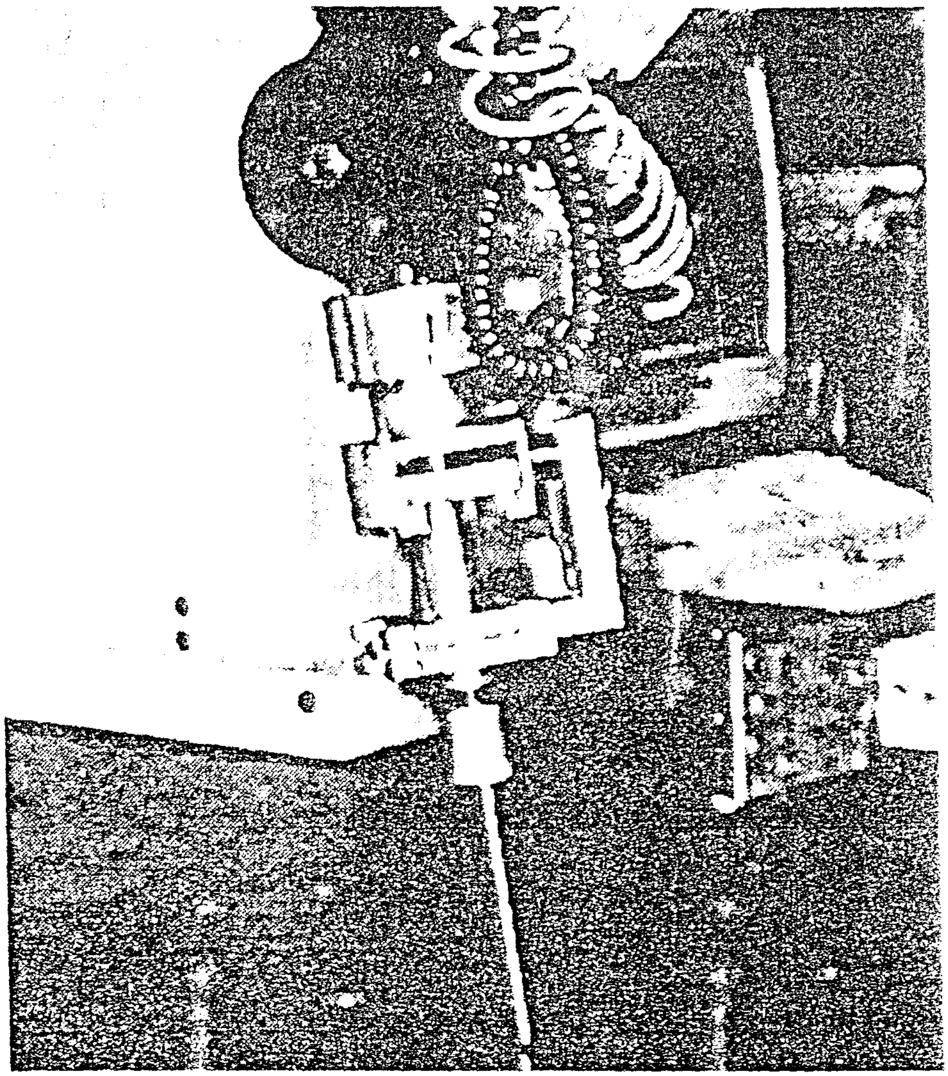


Figure 2. Robot touch constraint plane using a *Probe*

Identifying the Kinematics of Robots and their tasks

D.J. Bennet , J.H. Hollerbach

- Identifying both the manipulator kinematics and the task geometry = identifying the associated closed kinematic chain

- Using only joint angle readings and self motions without endpoint sensing

- Consistency conditions

- General Parameter identifiability conditions

1 Introduction

- Extends open-loop kinematic calibration methods to include the identification of the geometry (kinematics) of a task

- a door with a handle : two dof passive manipulator

- In order to identify environment, it must be moved

- manipulator # interacting with an environment generally forms a single loop closed kinematic chain
  ↳ 3 position and 3 orientation Eqs.

- ~~Eqs~~ Spatial Closed-Chain Eqs. 6개이므로
  unsensed passive dof 가 5개까지는 Closed-Chain이 움직일 수 있으므로 5 dof task kinematics는 identify 할 수 있다.

- Closed single-loop Kinematic Calibration

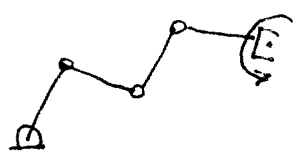
Summary

① the environment kinematics must be moved to be identified

② joint angle readings are enough requirements

③ identify both manipulator kinematics and enviroment kinematics

Specific Example



changing orientation

- manipulator kinematics + hinge의 position & orientation을 결정해야 한다.

- kinematic loop closure equation을 푸는 것은 problematic한 경우가 발생하므로 Jacobian의 column을 조사하여 어떤 parameters가 identifiable 한 지를 결정

~~○ 이 방법은~~

Historical Perspective

- 이 방법은 class of iterative open-loop kinematic calibration techniques

- Mechanism synthesis 와 비교. ~~Mecan~~

  ↳ 1 sensor로 전체 kinematics를 identification

  그러나 한 algorithm 6개 joint sensor 요구

2. Kinematic Model

2.1 Coordinate Convention

- DH Parameter (Paul's way)

$$A_j = Rot(z, \theta_j') \, Trans(z, s_j) \, Trans(x, a_j) \, Rot(x, \alpha_j)$$

$$T_c = \underline{\underline{A_0}} A_1 \cdots A_6$$

base coord. to 1st coordinate (fixed transformation)

~~Kinematic~~ geometric parameters (s_j, a_j, α_j)

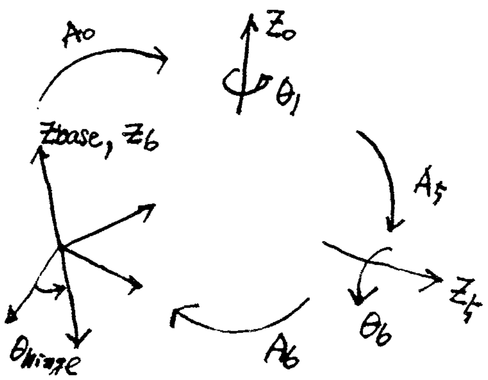
Nongeometric parameters relating the θ_j' to the joint angle measurements : Assumes that the only joint angle error is the offset θ_j^{off} $\rightarrow$ $\theta_j' = \theta_j + \theta_j^{off}$

2.2 Task Geometry

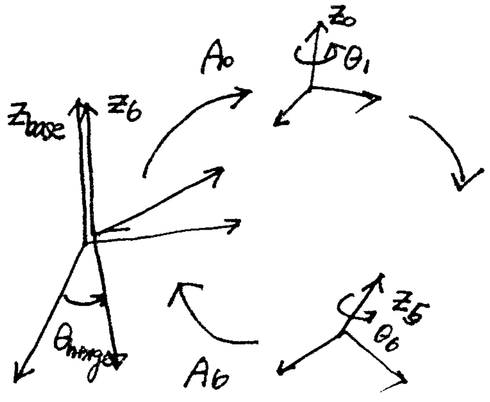
End point z axis를 도어의 힌지 axis와 일치시킨다.

base coord.의 z axis를 도어의 힌지 axis와 일치시킨다.

최종적으로 end point origin과 base frame origin을 동일점으로 가야 한다.



2.3 Closed loop Model



$$T_c = A_0 A_1 \cdots A_6 = \begin{bmatrix} R_c & dP_c \\ 0\,0\,0 & 1 \end{bmatrix}$$

Model Parameter의 오차가 없다면

$$R_c = \begin{bmatrix} \cos(\theta_{hinge}) & -\sin(\theta_{hinge}) & 0 \\ \sin(\theta_{hinge}) & \cos(\theta_{hinge}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$dP_c = [\,0 \quad 0 \quad 0\,]^T$$

그러나, ~~불~~ Model Parameter의 오차가 발생하여 $T_{base} \to T_c$ 는 위의 식과 다르게 된다.

Infinitesimal displacement $dP_c = [\,dx_c \; dy_c \; dz_c\,]^T$

$$R_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\delta x_c \\ 0 & \delta x_c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \delta y_c \\ 0 & 1 & 0 \\ -\delta y_c & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\delta x_c, \delta y_c$ are infinitesimal, θ_z is finite

$$R_c = \begin{bmatrix} 1 & 0 & \delta y_c \\ 0 & 1 & -\delta x_c \\ -\delta y_c & \delta x_c & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & -\sin\theta_z & 0 \\ \sin(\theta_z) & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & \delta y_c \\ \sin(\theta_z) & \cos(\theta_z) & -\delta x_c \\ -\delta y_c \cos(\theta_z) + \delta x_c \sin(\theta_z) & \delta y_c \sin\theta_z + \delta x_c \cos\theta_z & 1 \end{bmatrix}$$

앞에서

$$\theta_z = atan\left([R_c]_{2,1}/[R_c]_{1,1}\right)$$

∂x_c와 ∂y_c를 구하기 위하여

$$\begin{bmatrix} 1 & 0 & \partial y_c \\ 0 & 1 & -\partial x_c \\ -\partial y_c & \partial x_c & 1 \end{bmatrix} = \underline{R_c} \begin{bmatrix} \cos(\theta_z) & +\sin\theta_z & 0 \\ -\sin(\theta_z) & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus, the modeled end point location at ith joint configuration $\underline{\theta}^i$

$$\underline{x}_c^i = (dx_c, dy_c, dz_c, \partial x_c, \partial y_c)^T$$

and the five kinematic loop closure Eqs.

$$0 = x_c^i(\underline{\theta}^i + \underline{\theta}_{off}, \underline{s}, \underline{a}, \underline{d}) + \underbrace{\underline{n}^i}_{\text{modeling and measurement noise}} \qquad -①$$

Unknown parameters to be identified

$$\underline{\varphi} = (\underline{\theta}_{off}^T, \underline{s}^T, \underline{a}^T, \underline{d}^T)$$

2.4 Differential Kinematics

Eq. ①은 non-linear $\rightarrow$ needs to be linearlized about a nominal guess at the parameters $\underline{\varphi}^0$

θ_z를 포함시켜서 full six vector of position and orientation

$$\underline{X}_c^i = (dx_c, dy_c, dz_c, \partial x_c, \partial y_c, \theta_z)^T$$

$$\Delta x_c^i = \frac{\partial x_c^i}{\partial \theta}\Delta\theta + \frac{\partial x_c^i}{\partial \underline{s}}\Delta s + \frac{\partial x_c^i}{\partial a}\Delta a + \frac{\partial x_c^i}{\partial d}\Delta d + n^i$$

$$\Delta \underline{x}^i = (0,0,0,0,0,\text{unknown}) - \underline{x}_c^i$$

$$\Delta S = S - S^\circ \quad , \qquad \Delta \varphi = \varphi - \underline{\varphi}^\circ$$

$$C^i = \frac{\partial x_c^i}{\partial \varphi} = \left[\frac{\partial x_c^i}{\partial \underline{\theta}} \quad \frac{\partial x_c^i}{\partial \underline{d}} \quad \frac{\partial x_c^i}{\partial \underline{a}} \quad \frac{\partial x_c^i}{\partial \underline{S}} \right]$$

$$\Delta x^i = C^i \Delta \varphi + n'^i \quad \leftarrow 6\text{개 eqs.}$$

θ_z를 제외하면

$$\Delta x^i = (0,0,0,0,0) - x_c^i$$

$$\Delta x^i = C^i \Delta \varphi + n'^i$$

2.4.1 Jacobian Calculation

No Problem!

3. Iterative Parameter Identification

3.1 Data Collection

A series of m configurations of the actual mechanism

$\Rightarrow$ m set of joint angle measurements $\underline{\theta}^i$

$$\Delta \underline{X} = C \Delta \varphi + n'$$

where

$$\cancel{\Delta \underline{X} = C \Delta \varphi}$$

$$C = \begin{bmatrix} C^1 \\ \vdots \\ C^m \end{bmatrix} , \quad \Delta X = \begin{bmatrix} \Delta x^1 \\ \vdots \\ \Delta x^m \end{bmatrix}$$

3.2 Minimization

$$\min \quad E^2 = (\Delta X - C\Delta\varphi)^T(\Delta X - C\Delta\varphi) \quad : \text{least squares method}$$

$$\Delta\varphi = (C^TC)^{-1}C^T\Delta x \quad -①$$

$$\varphi_{i+1} = \varphi_i + \Delta\varphi \quad -②$$

①②를 Iteratively 적용하면, quadratic convergence를 얻을 수 있다 단 C^TC가 계속시 ~~ever~~ invertible 하여야 한다.

4. Identifiability

mechanism의 움직임을 통하여 얻어진 Data에 대하여 uniquely 결정하지 못하는 parameter들이 있을 수 있다.

rank(C)가 full rank가 아니면 C의 Column들 사이에 linear dependence가 있고 따라서 C^TC는 singular 하게 된다.

이와 같은 경우가 계산 중에 ~~각종~~ parameter에 의하여 발생할 수도 있고 실제 parameter에 의해서도 발생할 수 있다. 실제 parameter에서 발생된다면 그 parameter들은 unidentifiable 하다.

mechanism에서 하나 이상의 parameter가 unidentifiable 하면 그 mechanism은 unidentifiable 하다고 한다.

Necessary and Sufficient Cond. for Columns of $\bar{C}$ to be independent

$$\sum_{j=0}^{6} K_j x_j^i + G_j z_j^i = 0 \quad \text{for } \forall i \in [1 \cdots m] \Rightarrow K_j = G_j = 0 \qquad -(19)$$

~~위 조건에서 S:~~

위 조건을 위반하면 C의 Column들은 linearly dependent

하지만 위 조건만을 가지고는 length parameter에 의한 것인지

Angle parameter에 의한 것인지 판정하기 어렵다.

위 조건을 위반하는 각 경우에 대하여 살펴보자

각 경우에서 ~~expressively~~ ~~expressivly 표현하~~지 구체적으로 언급하지

않는다 하여도 실문제는 near singular situation 의 발생이다.

(1) Parameters that enter linearly

Kinematic closure eq.

$$\sum_j s_j z_{j-1}^i + a_j x_j^i = \underline{0} \quad - (20)$$

Eq. (20) 은 Eq.(19)를 위반, linear dependence of the length paramet

당연한 결과, angle에 의해서 결정된 geometry는 arbitrarily scaled

해결 방법은 하나의 nonzero link length를 1로 정하고 다른 link

length 들은 이것에 근거하여 결정함

(2) Immobility

- Joint i 가 Immobile 하다면, two consecutive joint coordinates are fixed relative to one another. 이것은 $x_i, z_i, x_{i+1}, z_{i+1}$ 이 constant linear relation을 갖게 된다. 따라서 관계된 link 들의 parameter 들은 unidentifiable

  이 경우, 가상 link를 정의함~~과~~으로써, 전체 mechanism은 identifiable

- the problem of immobile joint 에 적절히 대처하기 위해서는 immobility가 발생하는 ~~Condition 을~~ 경우를 예측하여야 한다.

  → immobility cond.

(3) Inherent Singularities in the mechanism

Joint 당 4개의 parameter로 정의할 필요없이 더 적은 수로 정의 될때 일반적인 general rule 을 정하기가 까다롭다.

예를 들면 four bar linkage 가 평행사변형을 형성한다면

$X_1 + X_3 = 0$ 식(19) 위반 a_1, a_3는 unidentifiable

다만, ~~a_1+a_3 은~~ 그것의 Sum은 identifiable

(4) Singularities related to the coordinate description

As a result of a poor choice of coordinate description,

Mechanism 내에 arbitrary parameter를 발생

예를 들면, Parallel axes $z_i - z_{i-1} = 0$

(19) 위반, S_i와 S_{i+1}를 각각 identify 할 수 없다.

~~따라서~~ 하지만, $S_{i+1} - S_i$는 알아 낼 수 있다.

이 경우를 해결하기 위하여 Hayati's Modified DH parameters

(5) Not sufficient excitation

어떤 joint가 전혀 움직이지 않았다면 그 관절에 대한 정보를

구할 수 없다. Category 2처럼 immobile 경우가 된다.

관심 사항은 어떤 Data Set이 가장 ~~좋은 Iden~~ 효과적 일까?
한 Identification Algorithm에

Optimal data set depends on the geometry of mechanism

Simulation study가 적절 → 참고: Brian Paper (4/3)

(6) Transient singularities

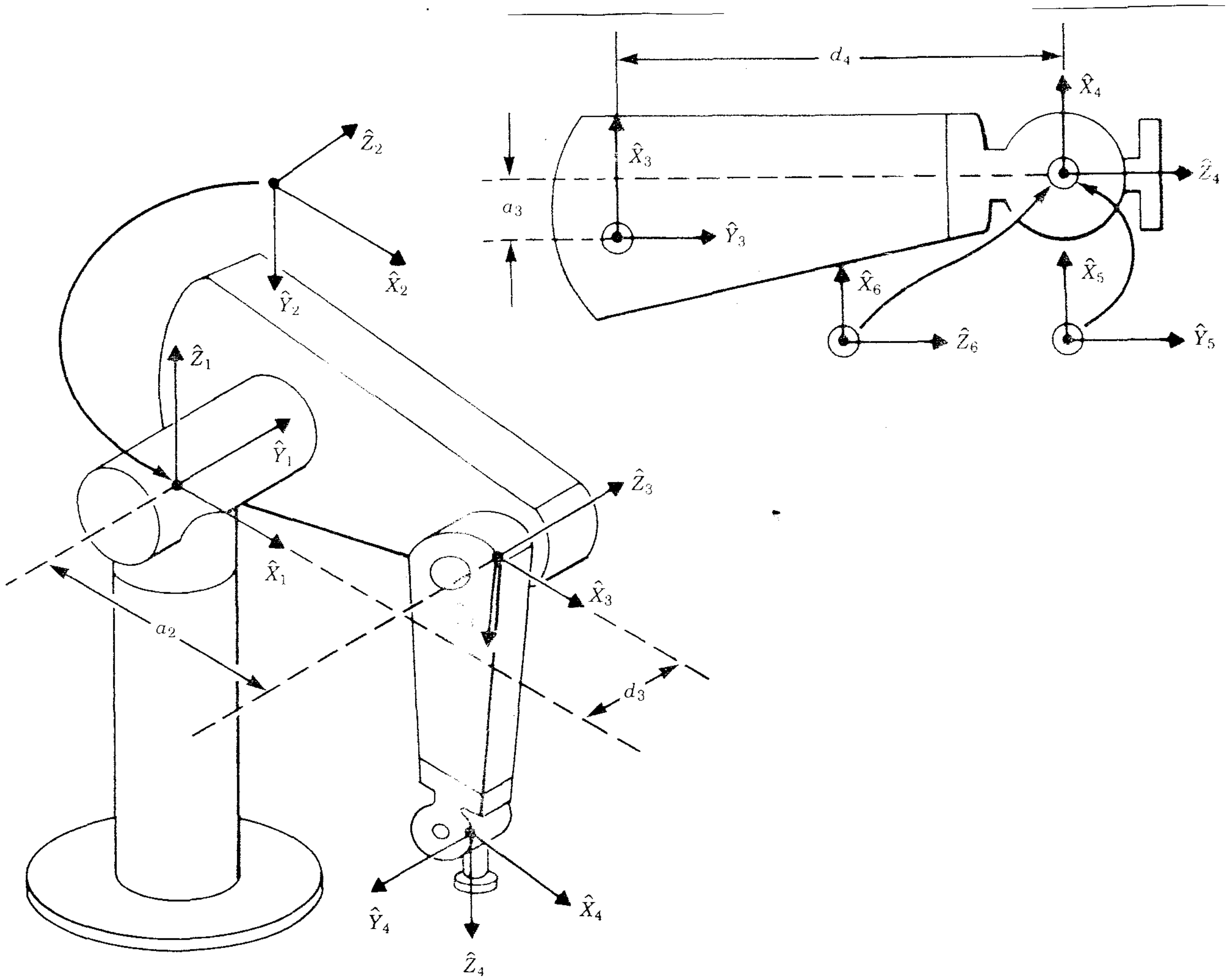
$$LM = E^2 + \lambda \Delta\varphi^T \Delta\varphi \quad \leftarrow E^2$$

$$\therefore \Delta\varphi = (C^T C + \lambda I)^{-1} C^T \Delta \underline{x}$$

Levenberg Marquardt algorithm

λ의 선정 (Kelman paper)

PUMA560 manipulator



Link parameters of the PUMA560

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	0	0	θ_2
3	0	0.4318	0.0203	θ_3
4	-90°	0.1254	0.4318	θ_4
5	90°	0	0	θ_5
6	-90°	0	0	θ_6

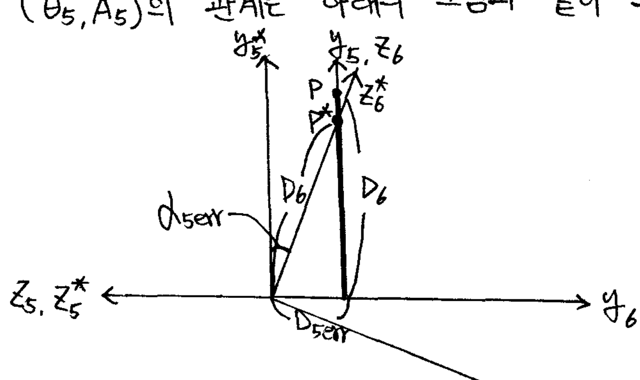
두개의 Point를 이용한 calibration

변수	RANK	상태	비고
A0	0	×	제외
A1	1	○	
A2	2	○	
A3	3	○	
A4	4	○	
A5	5	○	
D1	5	×	제외
D2	6	○	
D3	6	◎	제외
D4	7	○	
D5	8	○	
D6	9	◎	
α0	10	×	제외
α1	10	○	
α2	11	○	
α3	12	○	
α4	13	○	
α5	14	●	제외
θ1	14	×	
θ2	15	○	
θ3	16	×	제외
θ4	16	×	제외
θ5	16	●	
θ6	16	×	

○ : calibration가능 ● : 완전히 calibration되지 않는 경우
× : calibration불가 ◎ : 어떤 두개의 parameter가 일정한 비율로 변하는 경우

* 두개의 점에 대하여 각각 16개의 angle set을 구함
* A0, D1, α0, θ1은 기본적으로 calibration불가
* θ3, θ4는 angle set의 부족으로 인하여 calibration불가
* D2, D3는 두개의 합이 일정
* D6는 0.1로 고정
* θ6가 변하여도 다른 parameter에 영향을 주지 못함

* (α_5, D_5), (θ_5, A_5)의 관계는 아래의 그림과 같이 되어진다.



$P = (0, 0, D_6)$, $P^* = (0, 0, D_6 \cdot \cos(\alpha_{5err}))$ ⇒ Base에 대한 좌표

$D_{5err} = D_6 \cdot \sin(\alpha_{5err})$

이때 α_{5err}가 작은 값이되면 $P = P^*$가 된다.

1 Point(0.6 -0.4 0.3)에 대한 calibration

변수	상태	Rank	비고
A0	×	0	제외
A1	○	1	
A2	○	2	
A3	○	3	
A4	○	4	
A5	○	5	
D1	×	5	제외
D2	○	6	
D3	◎	6	D2와 D3의 합이 일정 : 제외
D4	○	7	
D5	○	8	
D6	◎	9	D6변화 : 다른 값이 비율적으로 변화 : 제외
$\alpha 0$	×	9	제외
$\alpha 1$	○	9	
$\alpha 2$	○	10	
$\alpha 3$	○	11	
$\alpha 4$	○	12	
$\alpha 5$	●	13	D5에 영향을 줌(수렴) : 제외
$\theta 1$	×	13	0이 아닌 다른값
$\theta 2$	○	14	
$\theta 3$	○	15	
$\theta 4$	○	16	
$\theta 5$	●	17→16	A5에 영향을 줌
$\theta 6$	×	17→16	

* 한점에 대하여 32개의 angle set을 구함

* D6는 0.1로 고정

1 Point(0.6 -0.4 0.3)에 대한 길이가 다른 두막대(0.1, 0.07)

변수	상태	Rank	비고
A0	×	0	제외
A1	○	1	
A2	○	2	
A3	○	3	
A4	○	4	
A5	○	5	
D1	×	5	제외
D2	○	6	
D3	◎	6	D2와 D3의 합이 일정 : 제외
D4	○	7	
D5	○	8	
D6	○	9	
$\alpha 0$	×	10	제외
$\alpha 1$	○	10	
$\alpha 2$	○	11	
$\alpha 3$	○	12	
$\alpha 4$	○	13	
$\alpha 5$	○	14	
$\theta 1$	×	15	제외
$\theta 2$	○	15	
$\theta 3$	○	16	
$\theta 4$	○	17	
$\theta 5$	○	18	
$\theta 6$	×	18	

* 길이가 다른 두막대에 대하여 각각 32개의 angle set을 구함
* 두막대의 길이를 알고 있어야함

Closed Loop

변수	상태	Rank	비고
A0	○	1	
A1	○	2	
A2	○	3	
A3	○	4	
A4	○	5	
A5	○	6	
A6	○	7	
D1	○	8	
D2	○	9	
D3	◎	9	D2와 D3의 합이 일정 : 제외
D4	○	10	
D5	○	11	
D6	◎	11	제외
D7	◎	11	제외
$\alpha 0$	○	12	
$\alpha 1$	○	13	
$\alpha 2$	○	14	
$\alpha 3$	○	15	
$\alpha 4$	○	16	
$\alpha 5$	○	17	
$\alpha 6$	○	18	
$\theta 1$	○	19	
$\theta 2$	○	20	
$\theta 3$	○	21	
$\theta 4$	○	22	
$\theta 5$	○	23	
$\theta 6$	○	24	
$\theta 7$	#		계산에 의해서 구해짐

* door hinge축과 base의 Z축을 일치 시켜야 한다.
* $\theta 7$은 계산에 의해서 구해지는 값이고 D7은 불필요한 data이다.
* A2, A3, D3, D4, D6중 하나는 값을 알고 있어야 한다.
* 여기서는 D6의 값을 0.1로 설정