Kinematic Calibration of Six-Axis Serial Robots

Friday, April 19, 2002

© ***John Hayes***

Department of Mechanical and Aerospace Engineering
Carleton University

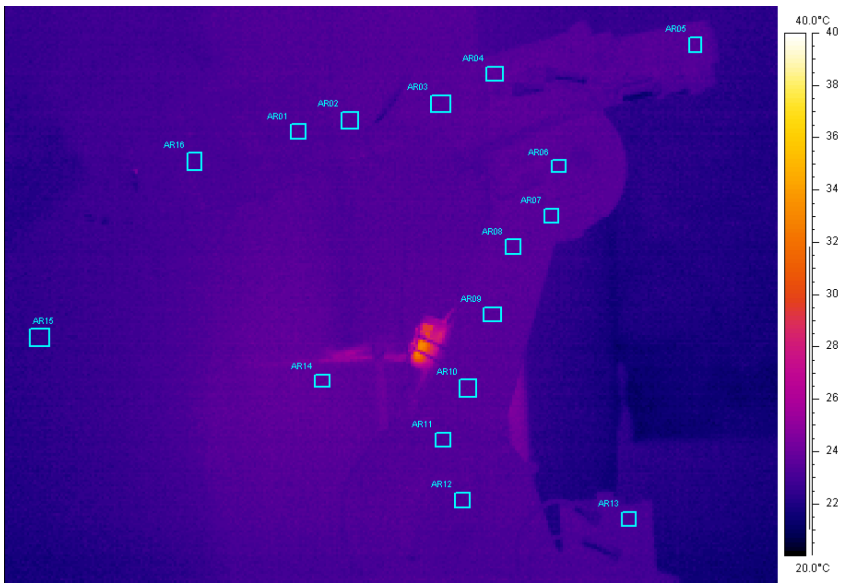
Carleton University
1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada
Tel: 613 520 2600 ext. 5661, FAX: 613 520 5715
email: jhayes@mae.carleton.ca
url:http://www.mae.carleton.ca/maehtmls/hayes.html

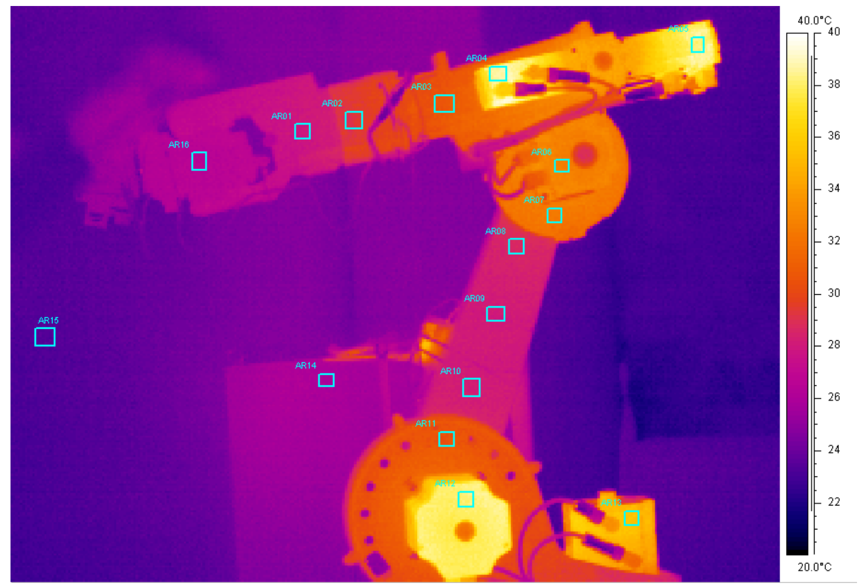
What Is Kinematic Calibration?

- Perhaps the most significant technical problems faced by a robot user are:
  - Issues involving off-line programming of the robot.
  - Sharing programs with other robots.
- Of all the issues the most complex is that of *kinematic calibration*.
- Large discrepancies between where the robot thinks it is and where it actually is usually exist. For positioning, a typical robot with a 1m reach will have $\pm$1mm variation.
- The reason for this variation is largely due to the details of the manufacture of the robot.

But, Sometimes Not...

- Infra-red image (cold).
- Infra-red image after 10 hours continuous motion sequence.





What Is Kinematic Calibration?

- The link lengths and joint angles are assigned nominal values in the controller. The accuracy of the robot depends on the variation between nominal and actual values.
- That is why calibration is needed.
- The errors induced by manufacture can be calibrated at a specific temperature by the manufacturer at the time of purchase of the robot.
- This is not a long term solution since wear from operating the robot introduces new errors.
- There are three main sources for these errors, and hence calibration procedures are classified according to the level of error they address.

What Is Kinematic Calibration?

- Joint level calibration:
  - This usually involves calibration of the drives and joint sensors.
- Kinematic model level calibration:
  - The purpose is to determine the actual kinematic geometry of the manipulator as well as correction for joint-angle errors. The model errors involve the changes in the physical dimensions of the links.
  - These errors arise from manufacturing tolerances, thermal expansion and system errors.
- Non-kinematic level calibration:
  - These errors are due to joint and link compliance, friction, clearance and deformations induced by dynamics.
- Our focus is on the kinematic model level. The errors at this level contribute about 95% of the overall positioning error.

What Is Kinematic Calibration?

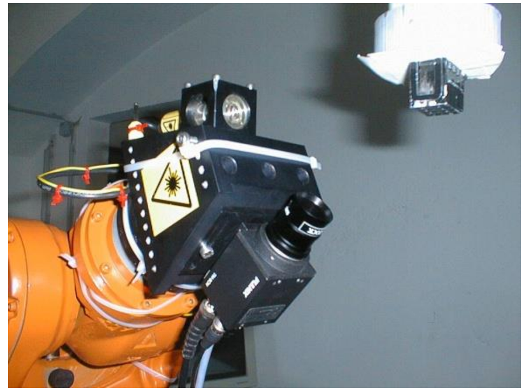
- Teaching (Taught Positions)
  - A *taught configuration* is one that the manipulator is moved to physically.
  - The joint position sensors are read and the joint angles stored.
  - When the robot is commanded to return to the taught configuration from some other, each joint is returned to the stored value.
- Repeatability
  - The *repeatability* of a robot is a measure of how precisely the robot can return to a taught configuration.
  - Repeatability has become the standard configuration performance indicator specified by manufacturers.
  - Typical values for articulated arms range from $\pm$0.1mm for the KUKA KR-15/2 to $\pm$0.035mm for the Stäubli RX130.

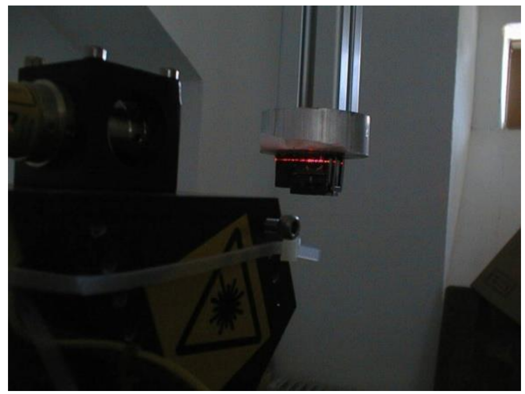
What Is Kinematic Calibration?

- Off-line programmed positions
  - This form of motion specification must be used when robotic tasks are guided by vision, or other motion guidance systems external to that of the robot.
  - The desired robot EE position and orientation is computed external to the robot controller.
  - The *inverse kinematics* of the robot must be computed in order to solve for the required joint angles.
- Accuracy
  - It may be that the computed pose is one that the robot has never before attained, hence repeatability is no longer sufficient to assess precision.
  - The *accuracy* of the robot is the precision with which a computed pose can be attained.
- The lower bound of *accuracy* is *repeatability*.

Implicit Calibration

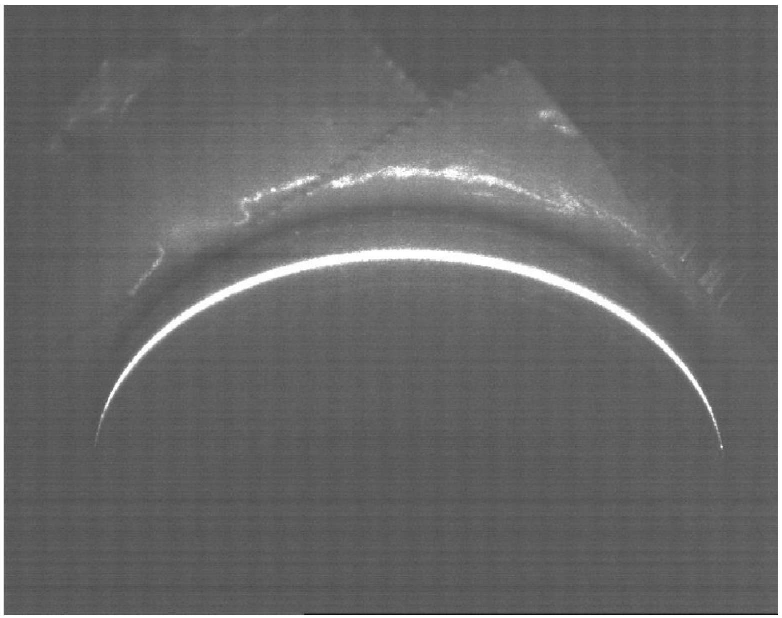
- Goal
  - Integrated optical/robotic measurement system
  - Application to rapid prototyping

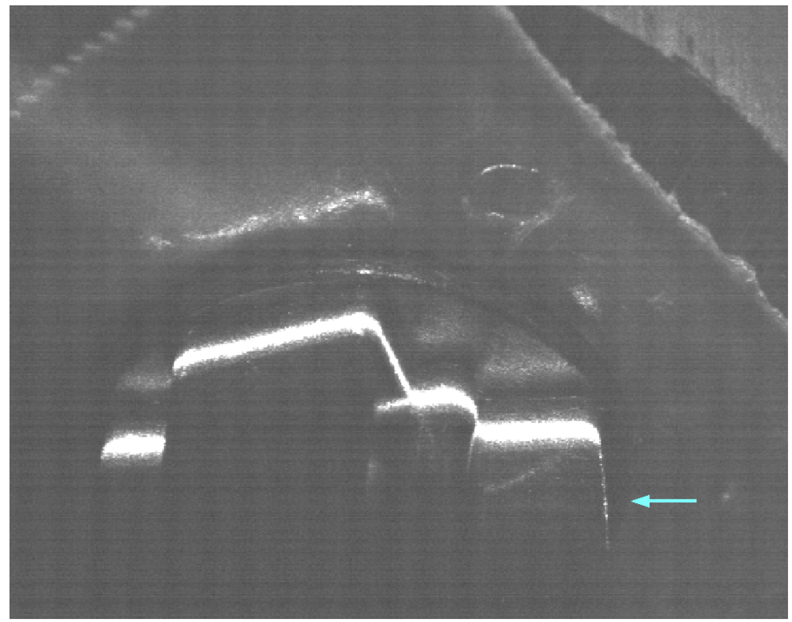




Implicit Calibration

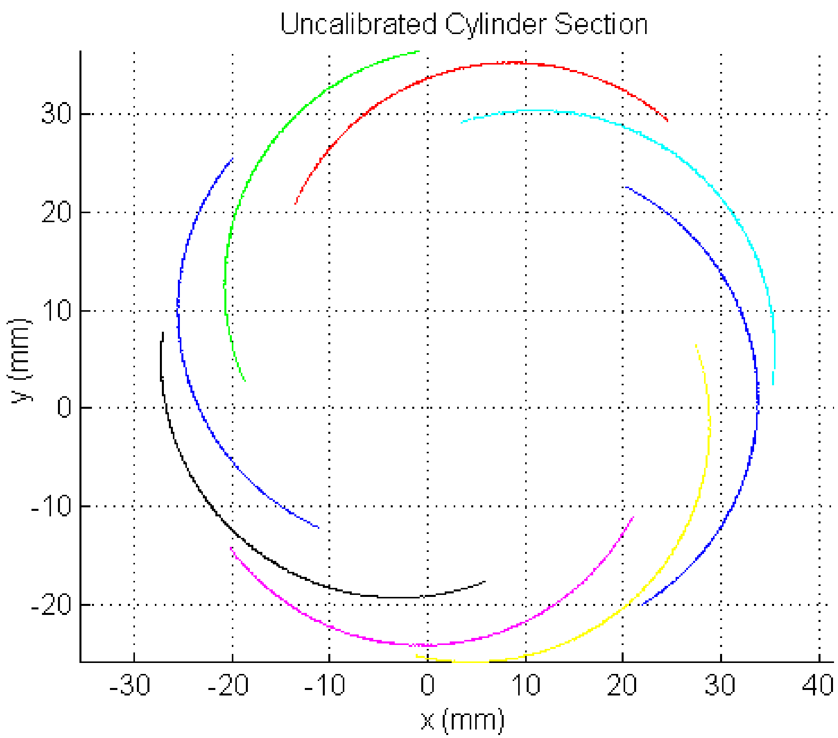
- Raw images of the calibration cylinder and connector:

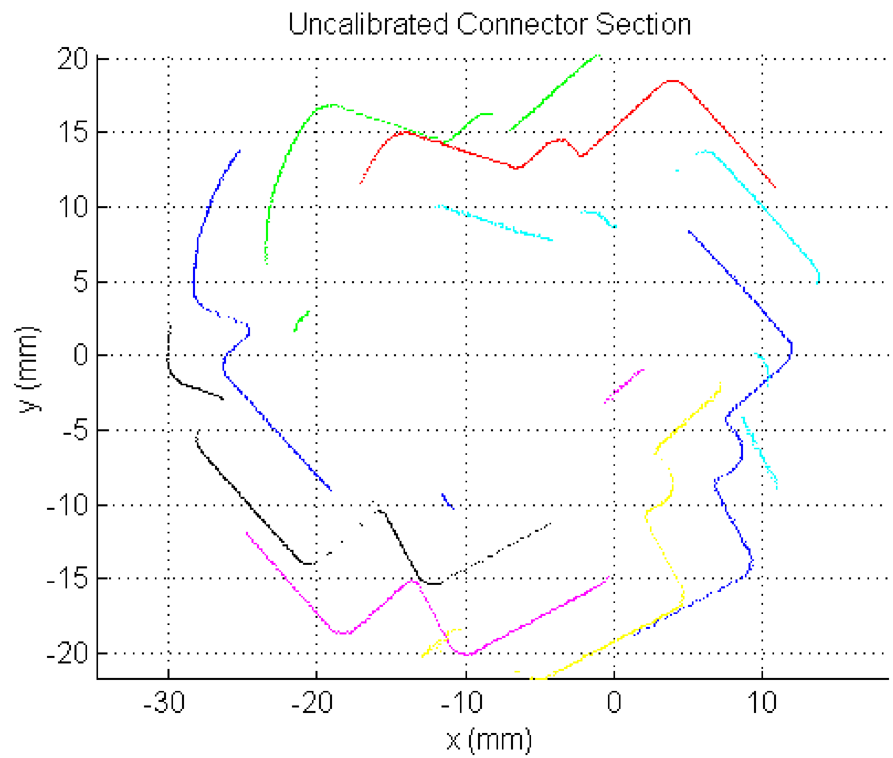




Implicit Calibration

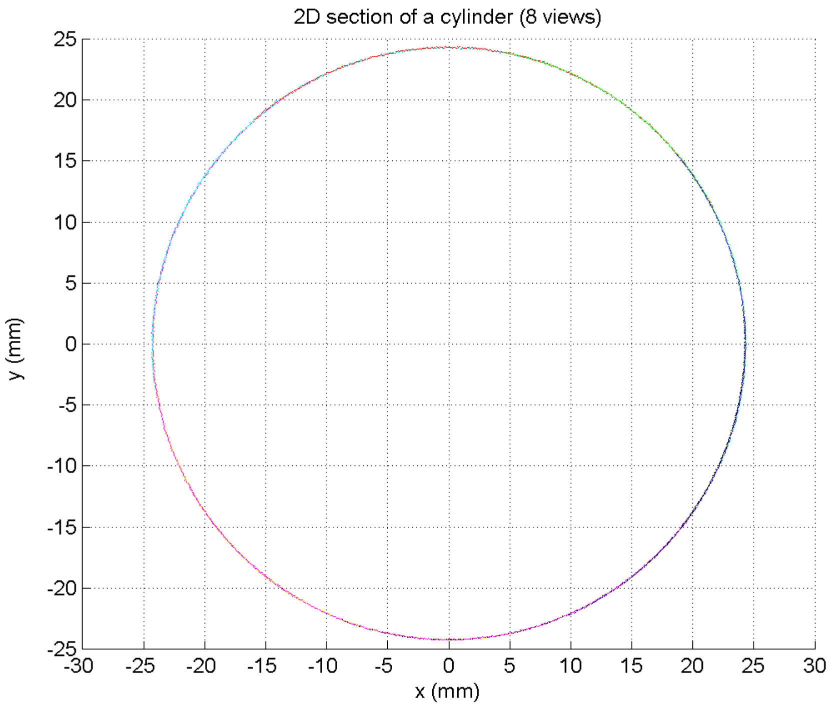
- Section raw data reconstruction without robot calibration:

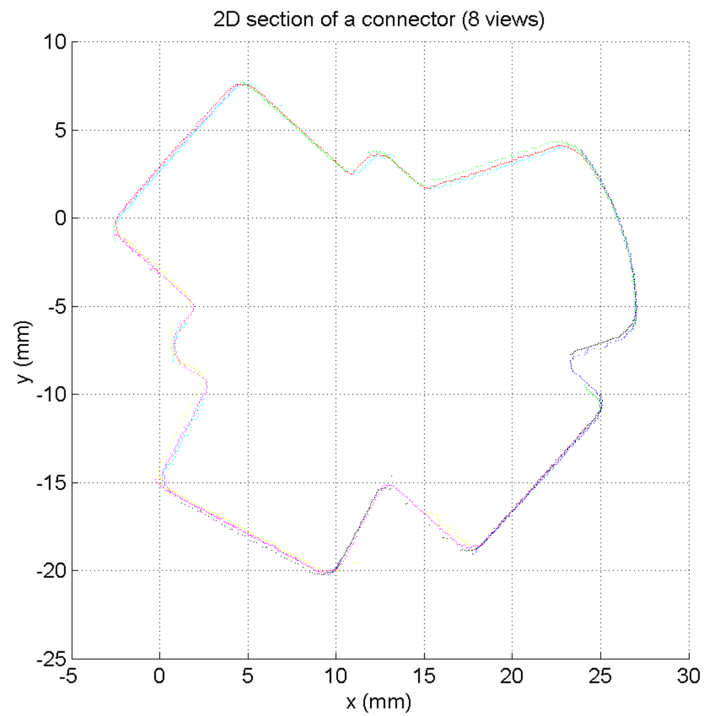




Implicit Calibration

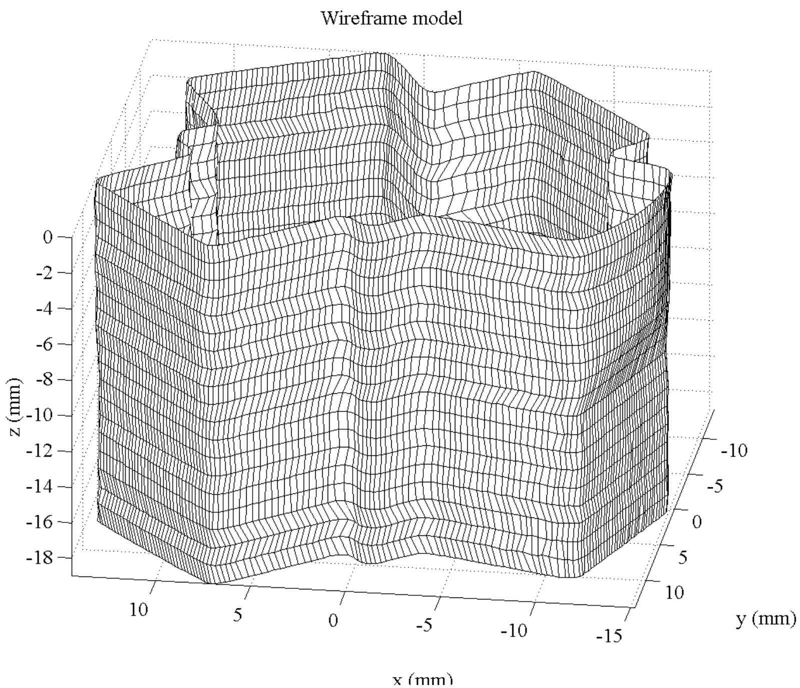
- Section raw data reconstruction after robot calibration:



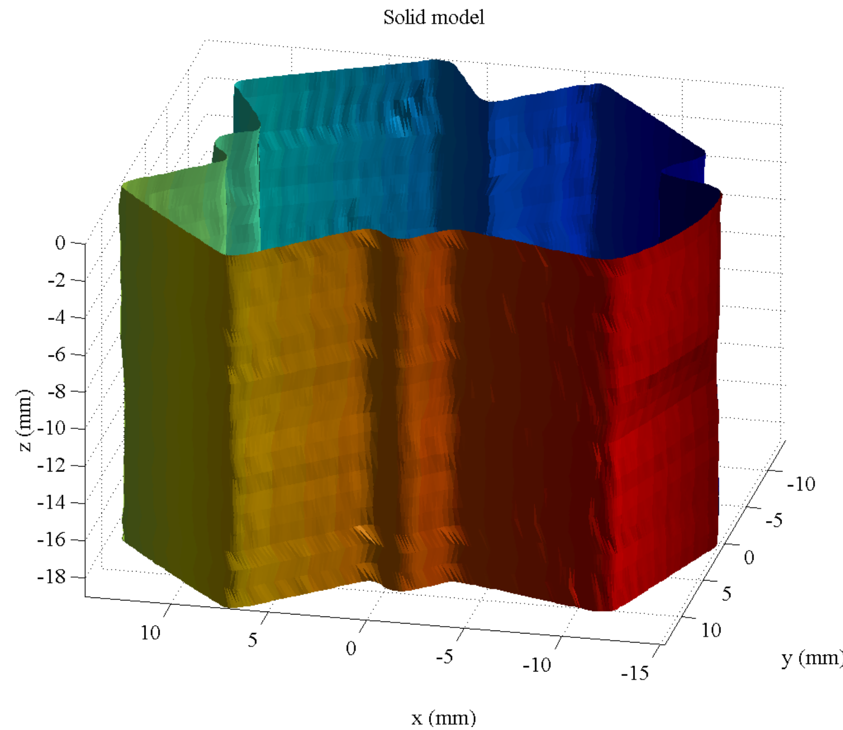


Implicit Calibration

- Solid model:



Wireframe model

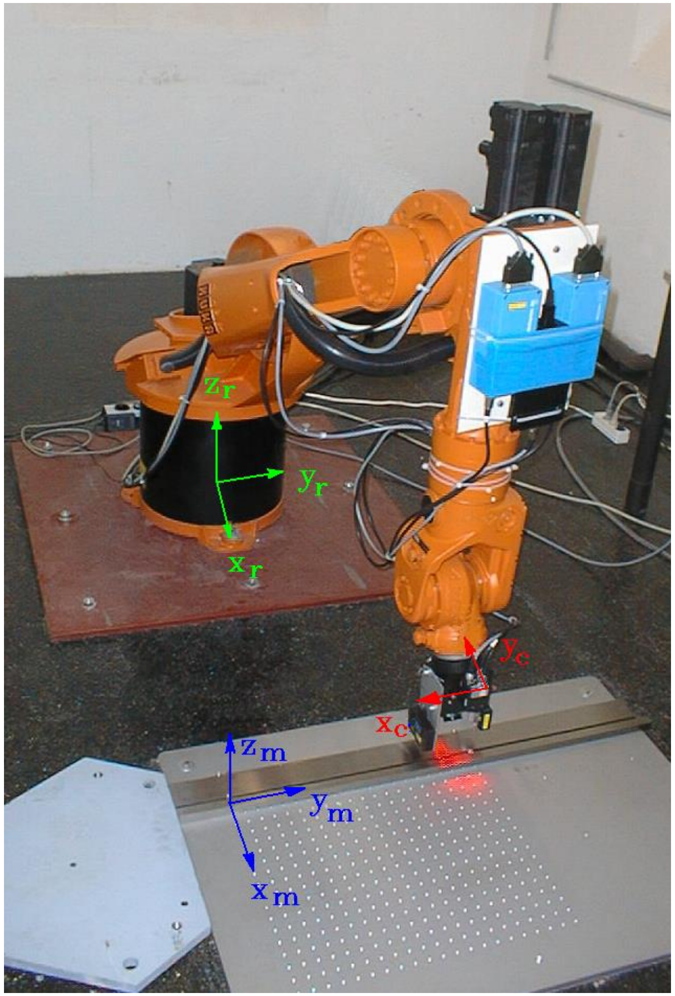


Rendered solid model

Relative Measurement Concept

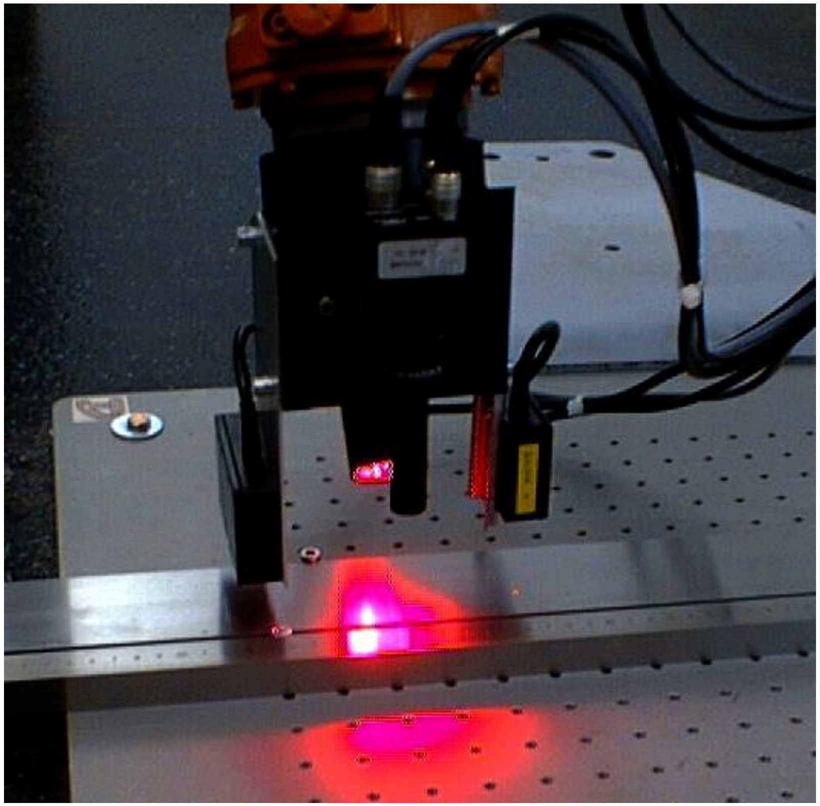
- Where the *robot* thinks it is: (x_r, y_r, z_r).
  - Output from the robot controller.
- Where the robot is *measured* to be:

  (x_m, y_m, z_m).
  - Output from our measurement sensors.
- Where LabVIEW *commanded* the robot to go: x_C, or y_C.
  - Distance in the x_r, or y_r directions the robot is commanded to move by LabVIEW.



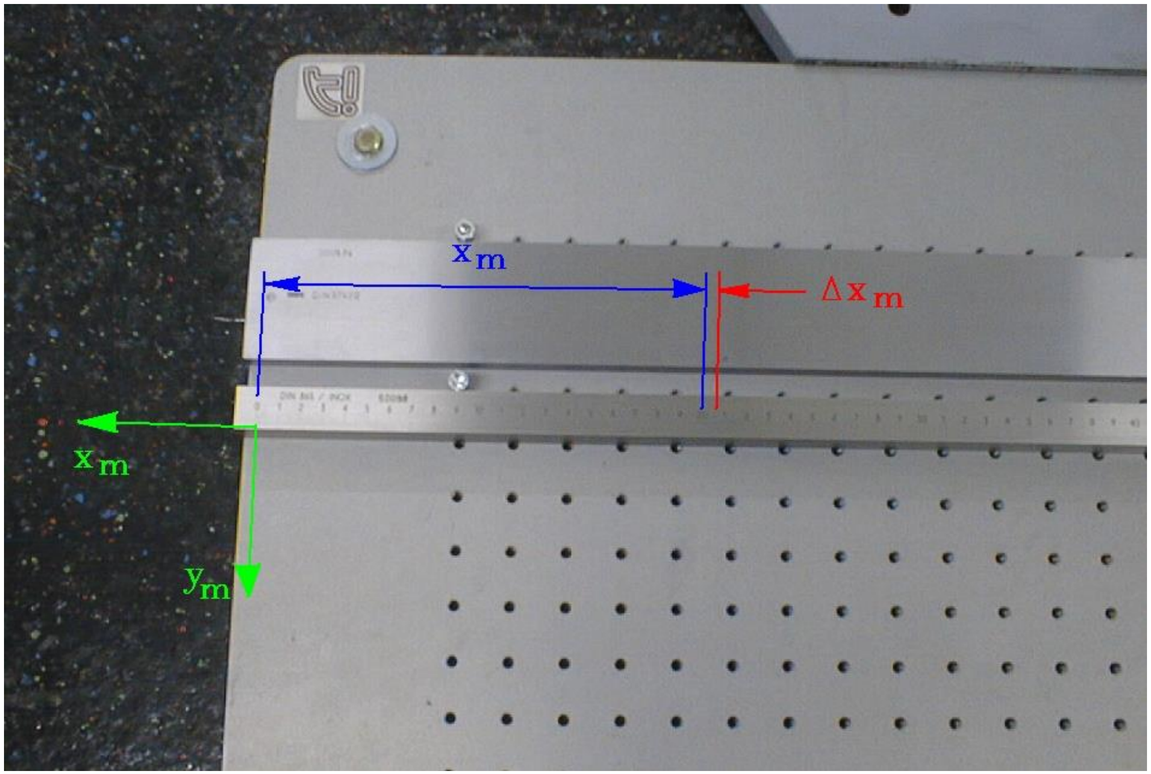
Relative Measurement Concept

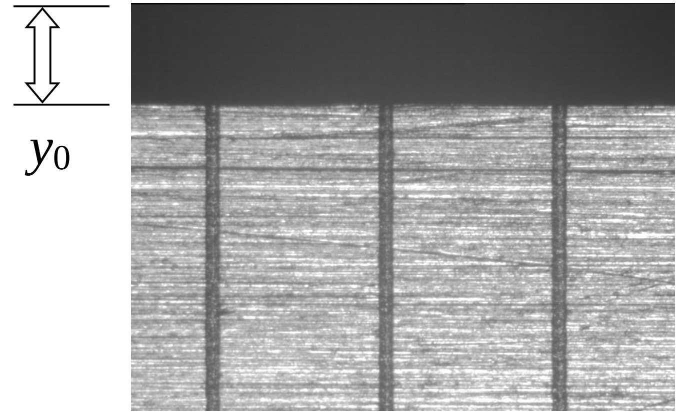
- Experiment
  - Measure linearly over the length of the ruled straight edge in 1cm increments along 2 distinct lines in the robot *x* and *y*-axis directions.
  - Record a CCD image of the ruler in each position.
  - The difference in the locations of the rulings in adjacent images is the difference between commanded and actual displacement in the direction of the ruler.
  - Change in z-axis direction obtained using a laser displacement sensor.



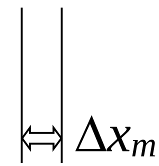
Relative Measurement Concept

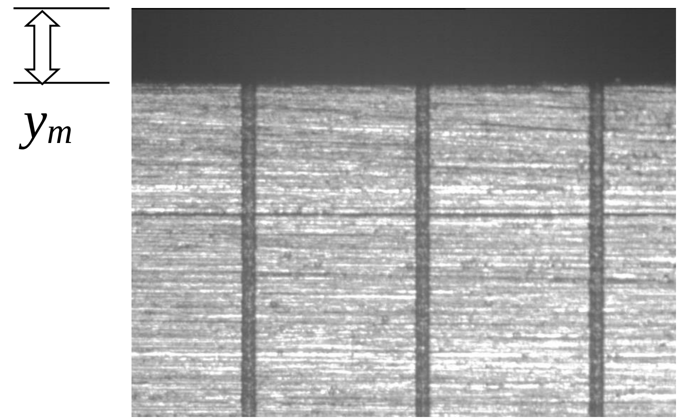
- Relative *X* and *Y* positioning.



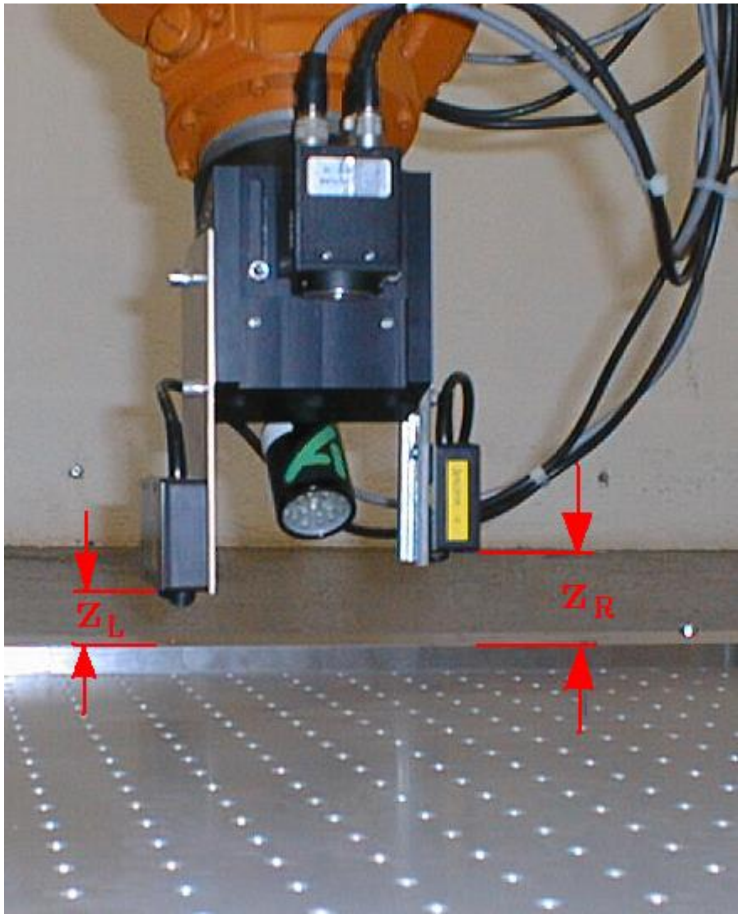


$$\Delta y_m = y_m - y_0$$





Relative Measurement Concept



- Relative *Z* positioning.
  - Two MEL sensors, left (M52L4) and right (M5L10).
  - Changes in the *Z* coordinate for motions along the length of the flat straight edge should be nil.
  - A linear Δ*Z* increment was added to each motion to compensate for misalignment of the straight edge relative to the *XY* base plane of the robot.

Kinematic Parameter Identification

- The forward kinematic (FK) relation between the kinematic model and the EE pose is given by:

$$x_i = \tilde{f}(\vartheta_i, \alpha, a, d) = \tilde{f}(\rho)$$

where $\vartheta_i, \alpha, a, d$ are vectors of the *Denavit-Hartenberg parameters* which describe the robot kinematic geometry.

- The vector $\boldsymbol{\rho}$ combines all kinematic parameters into one vector, $\boldsymbol{f}$ is a matrix function, and $\boldsymbol{x}$ is the vector of EE position and orientation (i.e. Euler angles).
- The calibration is based on iteration of the linearised FK.

Kinematic Parameter Identification

- The vector of joint angles is θ. Given that the procedure is linearised, we can model the joint characteristics linearly as

$$\vartheta_i = k\psi_i + \lambda$$

where ψ_i is the vector of joint angle sensor readings, λ is the vector of joint angle offsets and k the vector of joint angle gains.

- The first variation of the EE pose corresponding to variations in the DH parameters is given by:

$$\Delta x = \begin{bmatrix} \frac{\partial f}{\partial \vartheta_i} & \frac{\partial f}{\partial \alpha} & \frac{\partial f}{\partial a} & \frac{\partial f}{\partial d} \end{bmatrix} \begin{bmatrix} \Delta\vartheta_i \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix} = \begin{bmatrix} J_\vartheta & J_\alpha & J_a & J_d \end{bmatrix} \begin{bmatrix} \Delta\vartheta_i \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix} = J\Delta\rho$$

- Δx is the pose error, x_{meas}-$x_{\text{comp},}$, and $\Delta\rho$ is the vector of DH errors.

The Jacobian by Differentiation

- The Jacobian can be partitioned into linear and angular components.
- If only linear components are considered, as is the case when only linear measurements of the EE reference point are made, the sub-Jacobian relating the three EE reference point velocities to the six joint rates can be obtained by taking the partial derivatives of the transformation equation for the forward kinematics.
- If the DH kinematic model is used then in addition to the partial derivatives with respect to the joint angles, the partial derivatives with respect to the DH parameters are also required.

The Jacobian by Differentiation

- Taking the partial derivatives of *f(ρ)* with respect to all 24 parameters we obtain after eliminating the homogeneous coordinate (which vanishes upon differentiation):

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{\partial f}{\partial \rho} \begin{bmatrix} \dot{\vartheta} \\ \alpha \\ a \\ d \end{bmatrix}$$

The Jacobian by Differentiation

where

$$J = \frac{\partial f}{\partial \rho} = \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} \cdots \frac{\partial f_1}{\partial \phi_n} & \frac{\partial f_1}{\partial \alpha_1} \cdots \frac{\partial f_1}{\partial \alpha_n} & \frac{\partial f_1}{\partial \mathrm{a}_1} \cdots \frac{\partial f_1}{\partial \mathrm{a}_n} & \frac{\partial f_1}{\partial \mathrm{d}_1} \cdots \frac{\partial f_1}{\partial \mathrm{d}_n} \\ \frac{\partial f_2}{\partial \phi_1} \cdots \frac{\partial f_2}{\partial \phi_n} & \frac{\partial f_2}{\partial \alpha_1} \cdots \frac{\partial f_2}{\partial \alpha_n} & \frac{\partial f_2}{\partial \mathrm{a}_1} \cdots \frac{\partial f_2}{\partial \mathrm{a}_n} & \frac{\partial f_2}{\partial \mathrm{d}_1} \cdots \frac{\partial f_2}{\partial \mathrm{d}_n} \\ \frac{\partial f_3}{\partial \phi_1} \cdots \frac{\partial f_3}{\partial \phi_n} & \frac{\partial f_3}{\partial \alpha_1} \cdots \frac{\partial f_3}{\partial \alpha_n} & \frac{\partial f_3}{\partial \mathrm{a}_1} \cdots \frac{\partial f_3}{\partial \mathrm{a}_n} & \frac{\partial f_3}{\partial \mathrm{d}_1} \cdots \frac{\partial f_3}{\partial \mathrm{d}_n} \end{bmatrix}$$

- This 3x24 Jacobian is completely general and can be applied to any 6R wrist-partitioned robot architecture, but the terms are quite complicated.

The Jacobian by Differentiation

- If the displacement errors due to the difference between where the robot *thinks* it is and where it *actually* is are small relative to the link lengths then the velocity relation can be used to represent this difference.
- That is, the difference equations become

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \frac{\partial f}{\partial \rho} \begin{bmatrix} \Delta \varphi \\ \Delta \alpha \\ \Delta a \\ \Delta d \end{bmatrix} = J \Delta \rho$$

DH Parameter Identification

- We are interested in solving the previous equation for $\Delta\rho$.
- There are 24 unknowns and 3 equations.
- The system is underdetermined and has no solution in general.
- This suggests measuring more than one pose and approximating the solution in a least squares sense.
- Each pose creates a set of three more equations, which can be stacked for *m* poses.

$$\begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \\ \Delta x_m \end{bmatrix}_{3m\times 1} = \begin{bmatrix} J_1(\rho_1) \\ J_2(\rho_2) \\ J_3(\rho_3) \\ \vdots \\ J_m(\rho_m) \end{bmatrix}_{3m\times 24} \begin{bmatrix} \Delta\varphi \\ \Delta\alpha \\ \Delta a \\ \Delta d \end{bmatrix}_{24\times 1} = J_{3m\times 24}\Delta\rho$$

Calibration Procedure

- The calibration proceeds by iterating using SVD

$$\Delta\rho_i = J^{-1}\Delta x_{i-1}$$

where i indicates the iteration step, which gives updated DH parameters (at i=1, ρ_0 represents the nominal DH parameters).

$$\rho_i = \rho_{i-1} + \Delta\rho_i$$

- The computed pose is obtained by computing the FK

$$x_{i(\text{comp})} = f(\rho_{i-1} + \Delta\rho_i) = f(\rho_i)$$

Calibration Procedure

- In an absolute measurement based system each pose, x_{meas} is measured once, and is constant, while $x_{i(\text{comp})}$ is updated at each iteration.
- In the relative measurement based system the measurement data directly yields Δx_i.
- In an absolute measurement system, the Δx is computed as

$$\Delta x_i = x_{\text{meas}} - x_{i(\text{comp})}$$

- Knowing Δx_0 from the processed data provided by the vision and laser measurement system, and knowing the initial value of x_{comp} means the absolutely measured pose is embedded in the existing relatively measured data.

Calibration Procedure

- We can compute the constant value of x_{meas} from

$$x_{\text{meas}} = \Delta x_0 + x_{0(\text{comp})}$$

- For estimating the DH correction factors, $\Delta\rho$, we require x_{meas} so we can iteratively update $\Delta\rho_i$ until it is vanishingly small.
- Now we can use SVD to solve $\Delta\rho_i = J^{-1}\Delta x_{i-1}$.
- Each subsequent iteration provides a new $\Delta\rho_i$ until the convergence criterion, κ, is satisfied:

$$\left\| J\Delta\rho_{i+1} - \Delta x_i \right\| \leq \kappa$$

Calibration Procedure Summary

1. Move the robot through m+1 poses. Compute $x_{0(\text{comp})}$ and J using the nominal DH parameters and the FK.
2. Upon processing the measurement data compute x_{meas}, which is determined form the difference between adjacent poses.
3. Now the iterative parameter identification commences. First, compute $\Delta\rho_{i+1}$ using SVD.
4. Check if $\left\| J\Delta\rho_{i+1} - \Delta x_i \right\| \leq \kappa$, if not continue.
5. Update the DH parameters, J, $x_{(\text{comp})}$ and Δx.
6. Go to step 3.

Simulation

- A fictitious set of synthetic DH parameter errors was created:

$$\rho_{\text{actual}} = \rho_{\text{nominal}} + \Delta\rho_{\text{synthetic}}$$

- A perfect measurement system simulator then provided:

$$\Delta x = x_{\text{meas}} - x_{\text{comp}}$$

- A KUKA KR-15/2 robot was modeled using the nominal DH parameters in Table I.
- An arbitrary initial pose was selected and constant increments were sequentially added to the initial values to change the configuration, listed in Table II.
- The increments were selected so each joint would move in a different way, and after 100 poses the robot remained in the workspace.

Simulation

Table I
Nominal DH-parameters for Kuka KR-1512

Link *i*	θ_i	α_i **(deg)**	a_i **(mm)**	d_i **(mm)**
1	θ_1	90	300	675
2	θ_2	0	650	0
3	θ_3	90	155	0
4	θ_4	-90	0	600
5	θ_5	90	0	0
6	θ_6	0	0	140

Table II
Initial joint angles; constant increments

Initial position (deg)		Increments (deg)	
θ_1	0.0	$\Delta\theta_1$	-3.0
θ_2	-90.0	$\Delta\theta_2$	3.0
θ_3	0.0	$\Delta\theta_3$	-2.0
θ_4	0.0	$\Delta\theta_4$	-3.5
θ_5	0.0	$\Delta\theta_5$	3.2
θ_6	0.0	$\Delta\theta_6$	-2.5

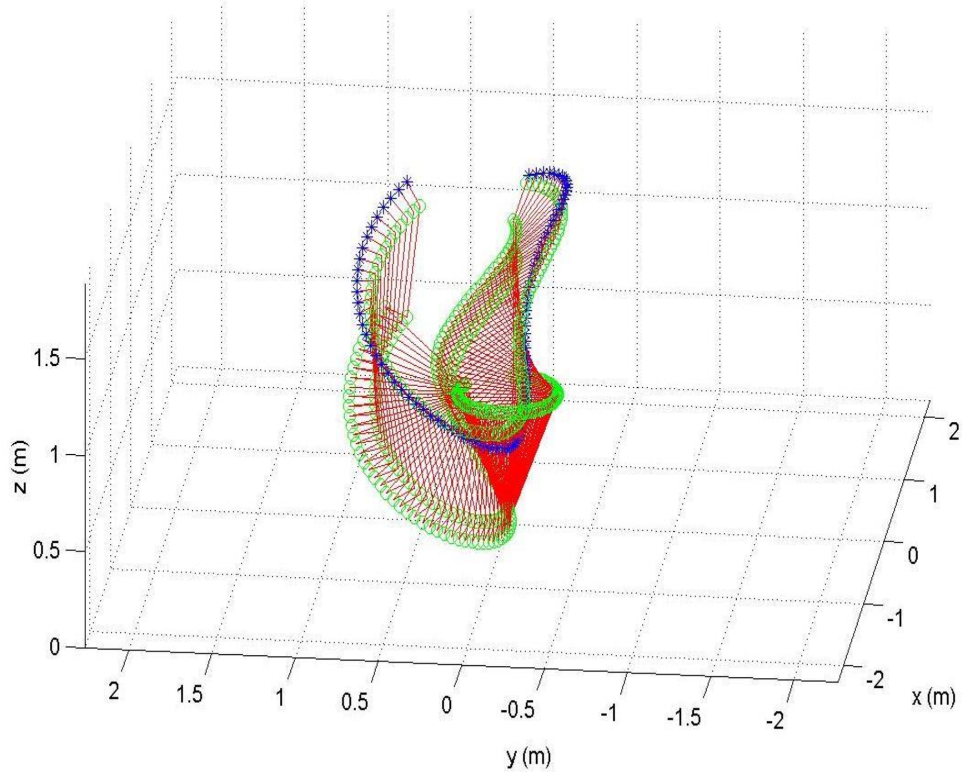
Simulation

- The s_i threshold was set to

$$\mathrm{rank}(V)\varepsilon \approx 5.33\times 10^{-15}$$

where rank(V)=24, and on the computer used ε=2.2204 x 10^{-16}

- The convergence criterion was set to be $\kappa=10^{-14}$.



The 100 robot poses.

* is the EE reference point

o are the joint centres

Results

- 1 singular value was annihilated after 3 iterations.
- 23 of 24 parameters errors were identified exactly.
- CPU time was 24.53 sec.

Results

DH-Error	Synthetic	Identified	Difference
Joint Angle	(rad)	(rad)	(%)
$\Delta\theta_1$	0.000870	0.000870	0.00%
$\Delta\theta_2$	0.000940	0.000940	0.00%
$\Delta\theta_3$	-0.001000	-0.001000	0.00%
$\Delta\theta_4$	0.000620	0.000620	0.00%
$\Delta\theta_5$	-0.000810	-0.000810	0.00%
$\Delta\theta_6$	0.000260	0.000000	100.00%
Twist error	(rad)	(rad)	(%)
$\Delta\alpha_1$	0.000157	0.000157	0.00%
$\Delta\alpha_2$	0.000130	0.000130	0.00%
$\Delta\alpha_3$	-0.000160	-0.000160	0.00%
$\Delta\alpha_4$	-0.000253	-0.000253	0.00%
$\Delta\alpha_5$	0.000462	0.000462	0.00%
$\Delta\alpha_6$	-0.000320	-0.000320	0.00%

DH-Parameter	Synthetic	Identified	Difference
Link length	(m)	(m)	(%)
Δa_1	0.000031	0.000031	0.00%
Δa_2	0.000051	0.000051	0.00%
Δa_3	0.000012	0.000012	0.00%
Δa_4	-0.000045	-0.000045	0.00%
Δa_5	0.000064	0.000064	0.00%
Δa_6	0.000058	0.000058	0.00%
Link offset	(m)	(m)	(%)
Δd_1	-0.000075	-0.000075	0.00%
Δd_2	0.000031	0.000031	0.00%
Δd_3	0.000022	0.000022	0.00%
Δd_4	0.000048	0.000048	0.00%
Δd_5	-0.000020	-0.000020	0.00%
Δd_6	0.000078	0.000078	0.00%

Results

- Recall the i^{th} column in V corresponding to the zeroed s_i gives the linear combination of $\Delta\rho$'s that is ill-determined.
- The s_i are arranged in decreasing order, hence s_{24} has been zeroed.
- Examining the 24^{th} column of V we see $V(6,24)=0.999...$
- This indicates $\Delta\rho_6$ is, by itself, not a trustworthy estimate.
- This is in agreement with the synthesized errors.

$$
V(:,24) = \left[\begin{array}{r}
\left.\begin{array}{r} -0.00000000000000 \\ -0.00000000000003 \\ 0.00000000000003 \\ 0.00000000000000 \\ -0.00000000000354 \\ 0.99999995592518 \end{array}\right\} \Delta\theta \\
\left.\begin{array}{r} -0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ -0.00000000000001 \\ 0.00028991673263 \end{array}\right\} \Delta\alpha \\
\left.\begin{array}{r} 0.00000000000000 \\ -0.00000000000000 \\ 0.00000000000000 \\ -0.00000000000000 \\ 0.00000000000121 \\ 0.00006401506558 \end{array}\right\} \Delta a \\
\left.\begin{array}{r} -0.00000000000000 \\ 0.00000000017731 \\ -0.00000000017731 \\ -0.00000000000000 \\ -0.00000000000000 \\ -0.00000001855904 \end{array}\right\} \Delta d
\end{array}\right]
$$