Computational Robotics

L2: Direct Geometric Model

Roberto Marino, PhD

Computational Modeling for Robotics and Human Computer Interaction
University of Messina - FCRLab
roberto.marino@unime.it

March 31, 2026



UniMe
1548

Why Do We Need Mathematical Models?

The design and control of a robot requires several **mathematical models**:

- **Geometric models** — location of the end-effector as a function of joint variables (and vice versa):
  - Direct Geometric Model (DGM)
  - Inverse Geometric Model (IGM)
- **Kinematic models** — end-effector velocity as a function of joint velocities (and vice versa).
- **Dynamic models** — relations between actuator torques/forces and joint positions, velocities and accelerations.

Approach

We use the **Khalil–Kleinfinger** notation, which provides a unified description for *all* mechanical articulated systems with a minimum number of parameters.

Serial Robot Structure

A serial robot consists of:

- $n+1$ **rigid links** (link 0 = base, link n = terminal)
- n **joints** (revolute **R** or prismatic **P**)

Joint variable q_j connects link $j-1$ to link j:

$$q_j = \bar{\sigma}_j \, \theta_j + \sigma_j \, r_j$$

with $\sigma_j = 0$ (revolute) or $\sigma_j = 1$ (prismatic).

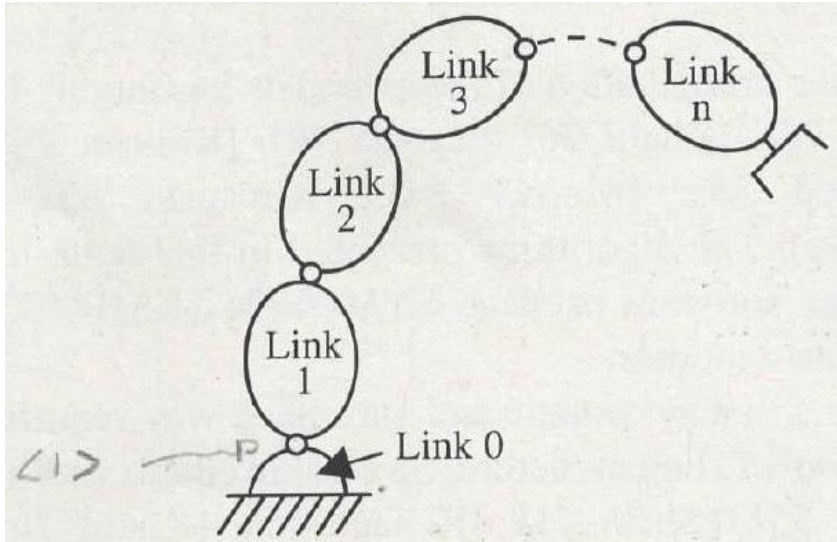


Fig. 3.1 – Robot with simple open structure

Frame Assignment Convention (Khalil–Kleinfinger)

A frame R_j is attached to each link j such that:

1. z_j axis is **along the axis of joint** j.
2. x_j axis is aligned with the **common normal** between z_j and z_{j+1}:
   - collinear axes: x_j any direction $\perp$ to them
   - parallel axes: x_j in the plane defined by them
   - intersecting axes: x_j normal to the plane at intersection
3. The **origin** O_j = intersection of x_j and z_j.
4. y_j completes a right-hand coordinate system.

Convention for base and terminal frames

R_0 is aligned with R_1 when $q_1 = 0 \Rightarrow \alpha_1 = 0$, $d_1 = 0$, $\bar{q}_1 = 0$. Similarly, x_n aligned with x_{n-1} when $q_n = 0 \Rightarrow \bar{q}_n = 0$.

The Four Geometric Parameters (MDH)

The transformation matrix beetween $j-1$ and j depends on **four parameters**:

Param.	Definition	Type
α_j	angle between z_{j-1} and z_j about x_{j-1}	const.
d_j	distance between z_{j-1} and z_j along x_{j-1}	const.
θ_j	angle between x_{j-1} and x_j about z_j	var. (R)
r_j	distance between x_{j-1} and x_j along z_j	var. (P)

The Four Geometric Parameters (MDH)

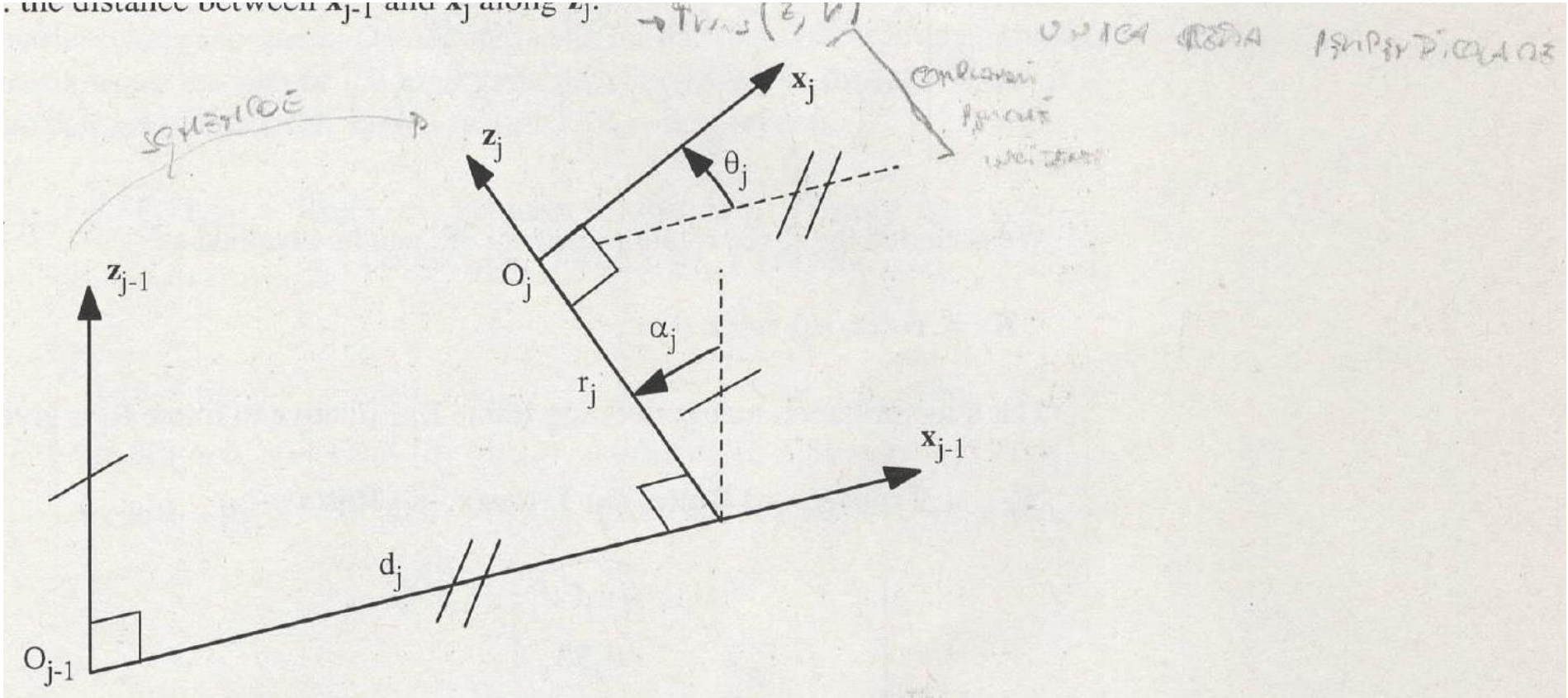


Fig. 3.2 – Geometric parameters (simple open structure)

MDH Transformation Matrix

The transformation from R_{j-1} to R_j is:

$$^{j-1}\mathrm{T}_j = \mathrm{Rot}(\mathrm{x}, \alpha_j)\ \mathrm{Trans}(\mathrm{x}, d_j)\ \mathrm{Rot}(\mathrm{z}, \theta_j)\ \mathrm{Trans}(\mathrm{z}, r_j)$$

$$^{j-1}\mathrm{T}_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

The rotation sub-matrix is:

$$^{j-1}\mathrm{R}_j = \mathrm{rot}(\mathrm{x}, \alpha_j)\ \mathrm{rot}(\mathrm{z}, \theta_j) \tag{3.3}$$

Inverse Transformation and Useful Notes

Inverse transformation ${}^{j}\mathrm{T}_{j-1}$:

$$ {}^{j}\mathrm{T}_{j-1} = \begin{bmatrix} {}^{j-1}\mathrm{R}_{j}^{T} & \begin{pmatrix} d_j C\theta_j \\ d_j S\theta_j \\ -r_j \end{pmatrix} \\ 0\ 0\ 0 & 1 \end{bmatrix} \tag{3.4} $$

Practical notes

- If $\mathrm{z}_j \parallel \mathrm{z}_{j+1}$: place x_j so that $r_j = 0$ or $r_{j+1} = 0$.
- If joint j is prismatic: place z_j so that $d_j = 0$ or $d_{j+1} = 0$.
- Joint encoders: $\mathrm{q} = \mathrm{K}\,\mathrm{q}_c + \mathrm{q}_0$.

Parallel Axes Simplification

When $\alpha_{j+1} = 0$ (i.e. $z_j \parallel z_{j+1}$), two consecutive matrices can be merged into one using *sum of joint variables*:

$$ {}^{j-1}\mathsf{T}_{j+1} = \begin{bmatrix} C(\theta_j+\theta_{j+1}) & -S(\theta_j+\theta_{j+1}) & 0 & d_j + d_{j+1}C\theta_j \\ C\alpha_j S(\theta_j+\theta_{j+1}) & C\alpha_j C(\theta_j+\theta_{j+1}) & -S\alpha_j & \cdots \\ S\alpha_j S(\theta_j+\theta_{j+1}) & S\alpha_j C(\theta_j+\theta_{j+1}) & C\alpha_j & \cdots \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5} $$

This **reduces the number of operations** significantly and will be exploited in the DGM examples.

Example 3.1 – Stäubli RX-90: Robot Overview

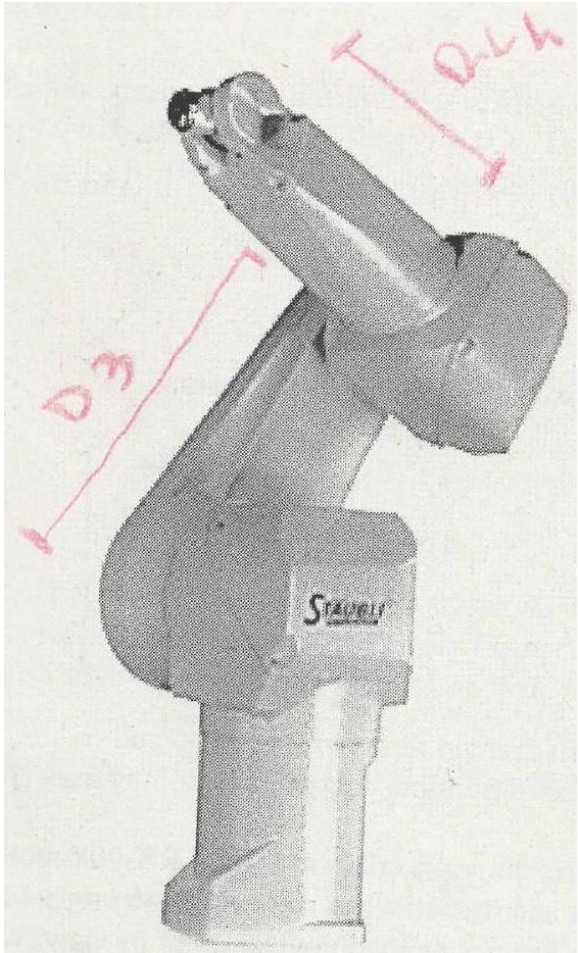


Fig. 3.3a – Stäubli RX-90

Architecture:

- **Shoulder**: RRR (joints 1–3)
- **Wrist**: 3R with intersecting axes (joints 4–6)

Methodology:

1. Place z_j axes on joint axes.
2. Place x_j axes by convention.
3. Determine $\alpha_j, d_j, \theta_j, r_j$ for each frame R_j/R_{j-1}.

Example 3.1 – Stäubli RX-90: Frames and Parameters

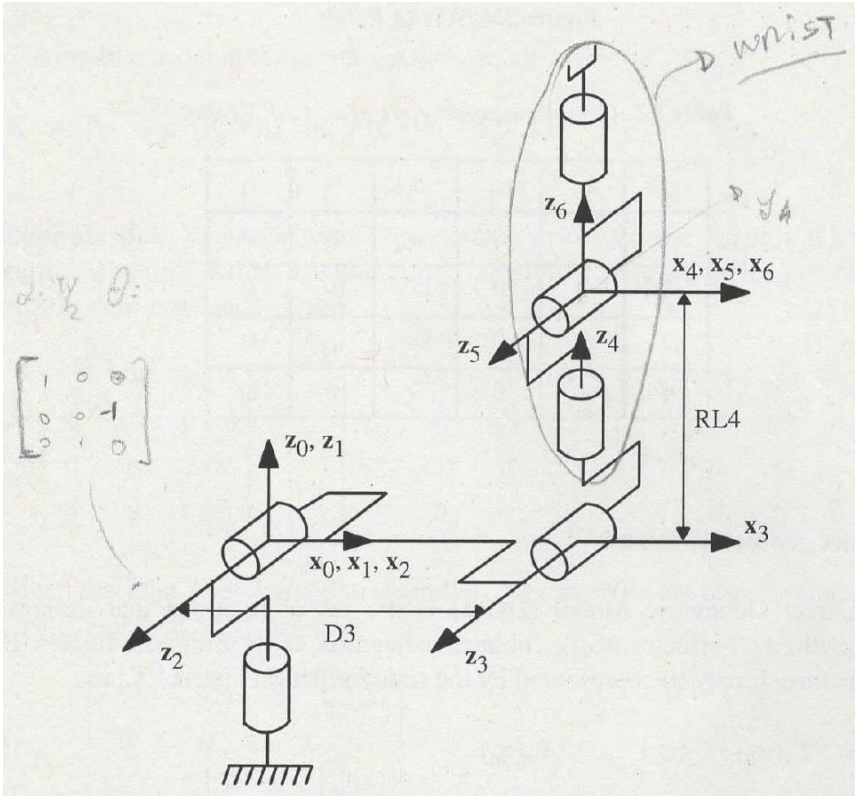


Fig. 3.3b – Link coordinate frames

j	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	$\pi/2$	0	θ_2	0
3	0	0	$D3$	θ_3	0
4	0	$-\pi/2$	0	θ_4	$RL4$
5	0	$\pi/2$	0	θ_5	0
6	0	$-\pi/2$	0	θ_6	0

Table 3.1 – Geometric parameters of the Stäubli RX-90

Example 3.2 – SCARA Robot

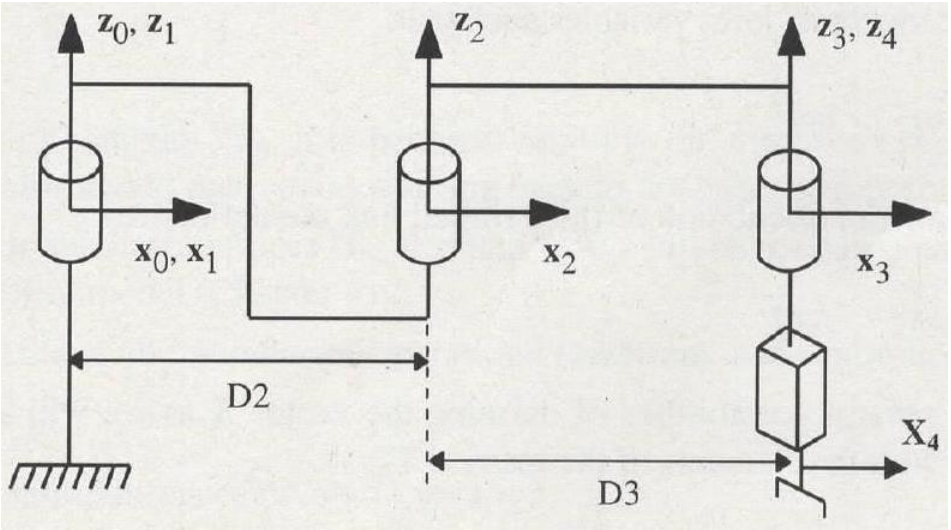


Fig. 3.4 – SCARA Robot (4-DOF)

j	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	0	$D2$	θ_2	0
3	0	0	$D3$	θ_3	0
4	1	0	0	0	r_4

Table 3.2 – Geometric parameters of SCARA

Note: joint 4 is **prismatic** ($\sigma_4 = 1$); joints 1–3 have parallel z axes $\Rightarrow$ can apply simplification (3.5).

Direct Geometric Model (DGM)

Definition

The **Direct Geometric Model** gives the location of the end-effector as a function of the joint coordinates $\mathrm{q} = [q_1, \ldots, q_n]^T$.

Matrix form – chain product of elementary transformations:

$$^0\mathrm{T}_n = {}^0\mathrm{T}_1(q_1)\,{}^1\mathrm{T}_2(q_2)\cdots{}^{n-1}\mathrm{T}_n(q_n) \tag{3.7}$$

Functional form:

$$\mathrm{X} = \mathrm{f}(\mathrm{q}) \tag{3.8}$$

where $\mathrm{X} \in \mathbb{R}^m$ collects the task-space coordinates of the end-effector (position + orientation).

End-Effector Pose Representation

Several choices exist for X. Using the columns of ${}^0\mathrm{T}_n$:

$$ {}^0\mathrm{T}_n = \begin{bmatrix} \mathrm{s} & \mathrm{n} & \mathrm{a} & \mathrm{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- **Full columns** ($m = 12$): $\mathrm{X} = [P_x, P_y, P_z, s_x, s_y, s_z, n_x, n_y, n_z, a_x, a_y, a_z]^T$
- **Reduced** ($m = 9$, since $\mathrm{s} = \mathrm{n} \times \mathrm{a}$): $\mathrm{X} = [P_x, P_y, P_z, n_x, n_y, n_z, a_x, a_y, a_z]^T$
- Other representations: Euler angles, RPY, quaternions, ...

Computational strategy

Multiply matrices **from end-effector to base** to obtain intermediate matrices $\mathrm{U}_j = {}^j\mathrm{T}_n$, which will also be used for the Inverse Geometric Model.

Example 3.3 – DGM of the Stäubli RX-90 (I)

Elementary matrices from Table 3.1 and eq. (3.2):

$$ {}^0\mathrm{T}_1 = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1\mathrm{T}_2 = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^2\mathrm{T}_3 = \begin{bmatrix} C3 & -S3 & 0 & D3 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Since $z_2 \parallel z_3$ (axes 2 and 3 parallel), apply eq. (3.5):

$$ {}^1\mathrm{T}_3 = \begin{bmatrix} C23 & -S23 & 0 & C2\,D3 \\ 0 & 0 & -1 & 0 \\ S23 & C23 & 0 & S2\,D3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

with $C23 = \cos(\theta_2+\theta_3)$, $S23 = \sin(\theta_2+\theta_3)$.

Example 3.3 – DGM of the Stäubli RX-90 (II)

Wrist matrices ($U_j = {}^{j}T_6$, computed right-to-left):

$$U_4 = {}^{4}T_6 = \begin{bmatrix} C5C6 & -C5S6 & -S5 & 0 \\ S6 & C6 & 0 & 0 \\ S5C6 & -S5S6 & C5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_3 = {}^{3}T_6 = \begin{bmatrix} C4C5C6-S4S6 & -C4C5S6-S4C6 & -C4S5 & 0 \\ S5C6 & -S5S6 & C5 & RL4 \\ -S4C5C6-C4S6 & S4C5S6-C4C6 & S4S5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 3.3 – DGM of the Stäubli RX-90 (III): Final Result

Final end-effector pose $U_0 = {}^0T_6 = {}^0T_1\, U_1$:

Orientation vectors:

$$
\begin{aligned}
s_x &= \\
&C1(C23(C4C5C6 - S4S6) - S23S5C6) - \\
&S1(S4C5C6 + C4S6) \\
s_y &= \\
&S1(C23(C4C5C6 - S4S6) - S23S5C6) + \\
&C1(S4C5C6 + C4S6) \\
s_z &= S23(C4C5C6 - S4S6) + C23S5C6 \\
a_x &= -C1(C23C4S5 + S23C5) + S1S4S5 \\
a_y &= -S1(C23C4S5 + S23C5) - C1S4S5 \\
a_z &= -S23C4S5 + C23C5
\end{aligned}
$$

Position vector:

$$
\begin{aligned}
P_x &= -C1(S23\, RL4 - C2\, D3) \\
P_y &= -S1(S23\, RL4 - C2\, D3) \\
P_z &= C23\, RL4 + S2\, D3
\end{aligned}
$$

Key observation

Position depends only on $\theta_1, \theta_2, \theta_3$ – typical *shoulder–wrist decoupling*!

Optimization of DGM Computation

Robot control requires **fast, real-time** computation of the DGM.

Symbolic customization reduces operations:

1. Expand matrix multiplications into **scalar equations**.
2. Replace non-trivial elements with **intermediate variables**.
3. **Eliminate** multiplications by 0 and 1, additions with 0.
4. Merge parallel-axes blocks using eq. (3.5): $C(\theta_j+\theta_{j+1})$ replaces two separate rotations.
5. Compute ${}^0s_n = {}^0n_n \times {}^0a_n$ (vector product) instead of a full matrix column.

Multiply right-to-left

The intermediate matrices $U_j = {}^jT_n$ (computed from the end-effector towards the base) are **also needed for the Inverse Geometric Model** (Chapter 4) $\Rightarrow$ zero extra cost.

Summary

Key Results

1. **Frame assignment** via the Khalil–Kleinfinger convention: four parameters $(\alpha_j, d_j, \theta_j, r_j)$ per link.
2. **MDH transformation matrix** ${}^{j-1}\mathrm{T}_j$ (eq. 3.2) provides the homogeneous transformation between consecutive frames.
3. **Parallel-axes simplification** (eq. 3.5) merges consecutive matrices using the sum of joint variables.
4. **DGM** = chain product ${}^{0}\mathrm{T}_n = \prod_j {}^{j-1}\mathrm{T}_j(q_j)$; computed right-to-left to produce intermediate matrices U_j.
5. Applied to the **Stäubli RX-90** (6-DOF) and the **SCARA** (4-DOF) robot.

Next Lecture

Inverse Geometric Model (IGM): how to find joint variables q from a desired end-effector pose X.