Technische Universität München **Chair of Automatic Control Engineering** Dr.-Ing. Marion Leibold **Chair of Information-Oriented Control** Dr.-Ing. Stefan Sosnowski	**Robot Control Laboratory** 1. Assignment

Excercise 1: Transformation

a) Write the following transformations in matrix notation and compare your results with the output of the defined Matlab routines from the *Robotics Toolbox* (in brackets).

1) Translation of 0.5 m in X direction $\boldsymbol{T}_1$ ($\mathbf{transl()}$)

2) Rotation of $90°$ around the Y axis $\boldsymbol{T}_2$ ($\mathbf{roty()}$)

3) Rotation of $-90°$ around the Z axis $\boldsymbol{T}_3$ ($\mathbf{rotz()}$)

b) The coordinate frames S_a and S_b have the following relation:

$$^{a}\boldsymbol{T}_b = \boldsymbol{T}_1 \cdot \boldsymbol{T}_2 \cdot \boldsymbol{T}_3$$

Compute the matrix $^{a}\boldsymbol{T}_b$. Compare it $^{a}T_b$ with $^{a}\boldsymbol{T}_{b1} = \boldsymbol{T}_3 \cdot \boldsymbol{T}_2 \cdot \boldsymbol{T}_1$. Discuss the commutativity of the multiplication of transformation matrices.

c) Given is the point $_{b}\boldsymbol{p}$ with respect to the coordinate frame S_b. Implement the function **cal_ap** to determine $_{a}\boldsymbol{p}$ with respect to coordinate frame S_a.

(4) $_{a}\boldsymbol{p}$ = cal_ap($_{b}\boldsymbol{p}$)	MAT
Input:	$_{b}\boldsymbol{p}$ 3×1 Vector
Output:	$_{a}\boldsymbol{p}$ 3×1 Vector
Allowed routines:	rotx(), roty(), rotz(), transl()

Next the same transformation matrix will be computed using different concepts.

d) Compute the RPY angles $\boldsymbol{\Omega}_1 = [\alpha, \beta, \gamma]$ from the transformation matrix $^{a}\boldsymbol{T}_b$ from b) with the routine $\mathbf{tr2rpy()}$. Please add the option $'zyx'$ to the routine $\mathbf{tr2rpy()}$ to get the same definition as in the lecture.

e) Implement the routine **cal_aTb2** that computes the homogeneous transformation matrix $^{a}\boldsymbol{T}_{b2}$ using the RPY angles and a translational vector as inputs. Use the equation on pages 2-7 of the material.

(4) $^{a}\boldsymbol{T}_{b2}$ = cal_aTb2($\alpha, \beta, \gamma, \boldsymbol{p}_{Sb}$)	MAT
Input:	α, β, γ Scalar angle in radians $\boldsymbol{p}_{Sb}$ 3×1 Vector
Output:	$^{a}\boldsymbol{T}_{b2}$ 4×4 homogeneous transformation matrix
Allowed routines:	rotx(), roty(), rotz(), transl()

Test: Compute the transformation matrix ${}^{a}\boldsymbol{T}_{b2}$ using the RPY angles Ω_1 from d) and the translational vector from a1) with your function. Additionally compute ${}^{a}\boldsymbol{T}_{b3}$ using the routine $\mathbf{rpy2tr()}$. Compare ${}^{a}\boldsymbol{T}_{b2}$, ${}^{a}\boldsymbol{T}_{b3}$ with ${}^{a}\boldsymbol{T}_{b}$.

f) Compute the Euler angles $\boldsymbol{\Omega_2} = [\phi, \theta, \psi]$ from the transformation matrix ${}^{a}\boldsymbol{T}_{b}$ from b) using the routine $\mathbf{tr2eul()}$.

g) Implement the function **cal_aTb4** that computes thenhomogeneous transformation matrix ${}^{a}\boldsymbol{T}_{b4}$ using Euler angles and a translational vector. For that task use the defitinion of Euler angles from the *Robotics Toolbox*:

$$^{a}\boldsymbol{R}_b(\phi, \theta, \psi) = \boldsymbol{R}(z, \phi) \cdot \boldsymbol{R}(y, \theta) \cdot \boldsymbol{R}(z, \psi)$$

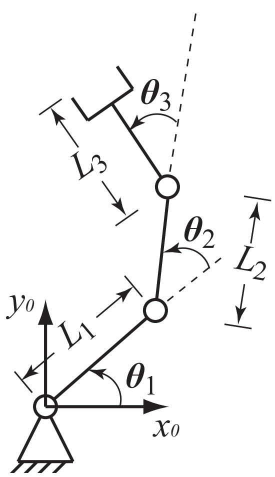
Achtung:
This defineition differs from the definition on pages 2-8 in the lecture material *Robotics Toolbox*.

(4)	${}^{a}\boldsymbol{T}_{b4} = \text{cal\_aTb4}(\phi, \theta, \psi, \boldsymbol{p}_{Sb})$	MAT
Input:	ϕ, θ, ψ scalar angles in radians	
	$\boldsymbol{p}_{Sb}$ 3×1 Vector	
Output:	${}^{a}\boldsymbol{T}_{b4}$ 4×4 homogeneous transformation matrix	
Allowed routines:	rotx(), roty(), rotz(), transl()	

Test: Compute the homogeneous transformation matrix ${}^{a}\boldsymbol{T}_{b5}$ using the Euler angles Ω_1 from d) and the translational vector from a1) with your function. Additional compute the transformation matrix ${}^{a}\boldsymbol{T}_{b5}$ using the routine $\mathbf{eul2tr()}$. Compare ${}^{a}\boldsymbol{T}_{b4}$, ${}^{a}\boldsymbol{T}_{b5}$ with ${}^{a}\boldsymbol{T}_{b}$.

Excercise 2: Denavit Hartenberg Convention and Forward Solution

Given is a planar robot with $N = 3$ segments, shown in figure (as in figure V.9 in the lecture material). The robot consists of three rotational joints θ_1, θ_2 and θ_3. The length of the segments is defined as $L_1 = 4\,\mathrm{m}$, $L_2 = 3\,\mathrm{m}$ and $L_3 = 2\,\mathrm{m}$.



a) Define the missing coordinate frames and the generalized rotation angles for the robot according to the Denavit Hartenberg Convention and write down the kinematic parameters in a table.

b) Compute manually ${}^0\boldsymbol{T}_E$ using homogeneous transformation matrices and extract the psoition of the end effector r_x, r_y and its orientation ϕ. The orientation ϕ is defined as the angle between the x_E– and the x_0 axis.

Hints:
$sin(\alpha + \beta) = sin(\alpha)cos(\beta) + cos(\alpha)sin(\beta)$
$cos(\alpha + \beta) = cos(\alpha)cos(\beta) - sin(\alpha)sin(\beta)$
$sin(\alpha - \beta) = sin(\alpha)cos(\beta) - cos(\alpha)sin(\beta)$
$cos(\alpha - \beta) = cos(\alpha)cos(\beta) + sin(\alpha)sin(\beta)$

Verification: Use the *Symbolic Math Toolbox for Matlab*, to derive the forward kinematics of the robot ${}^0\boldsymbol{T}_E(\theta_1, \theta_2, \theta_3)$ symbolically. You need to define symbols for the joint angles using $\mathbf{syms}$. Compare your results.

c) Implement the function **vwl** to compute the forward kinematics for the robot. Additionally the tansformation matrix should be in the output.

(10)	$[{}^0\boldsymbol{T}_E, r_x, r_y, \phi] = \text{vwl}(\theta_1, \theta_2, \theta_3)$	
Input:		$\theta_1, \theta_2, \theta_3$ scalar angles in radians
Output:		${}^0\boldsymbol{T}_E$ homogeneous transformation matrix
		r_x, r_y Position in x and y direction of the end effector
		ϕ Orientation of the end effector
Allowed routines:		rotx(), roty(), rotz(), transl()

Test: Test your results from excercises a)-c) using the routines $\mathbf{link}()$, $\mathbf{robot}()$ and $\mathbf{fkine}()$ from *Robotics Toolbox for Matlab*. Test the transformation ${}^0\boldsymbol{T}_E$ with the following joint angles:

1) $\Theta = (\theta_1, \theta_2, \theta_3)^T = (0°, 0°, 0°)^T$.

2) $\Theta = (\theta_1, \theta_2, \theta_3)^T = (10°, 20°, 30°)^T$.

d) Implement the function **cal_ss**. It should compute the boundaries of the work space of the platform and output 2D points that can be plotted.

(4) $\boldsymbol{P} = \mathrm{cal\_ss}()$	
Input:	—
Output:	$\boldsymbol{P} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & & y_n \end{bmatrix}$ Matrix consisting of 2D points $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$ approx.100 entries
Erlaubte Funktionen:	rotx(), roty(), rotz(), transl()

Test: Draw the workspace of the robot in the x-y plane. Furthermore you can animate the robot using $\mathbf{drivebot()}$. Additionally you should vary the joint angles.