Functions for the module Robotik

Get Started:

1. Download the robotic.tns file.
2. Open the robotic.tns file with the TI-Nspire™ CX CAS Student Software.
3. Connect your TI-Nspire™ CX CAS over the USB cable with your PC.
4. In the software go to File/Save to Handheld…
5. Double click on your TI-Nspire™ CX CAS in the appeared window.
6. Go to "MyLib", rename the file to "robotic" and press Save.
7. Open a new Calculator page on your TI-Nspire™ CX CAS.
8. Press the "doc" button.
9. Update the libraries by pressing the number 6.
10. To access the new functions, press the library button, the number 6 and search for "robotic".

Download this function documentation as pdf.

Functions:

robotic/atan2(y,x)

Function to calculate the arctan2. See Wikipedia

Parameters:

- y: sinus
- x: cosinus

Returns:

- Related angle θ

Note: Works for rad and deg. Calculater settings are crucial.

---

robotic/rotx(θ)

Function to get the rotation matrix around the x-axis.

Parameters:

- θ: Angle around the x-axis. Works for rad and deg. Calculater settings are crucial.

Returns:

- rotation matrix (4x4).

---

robotic/roty(θ)

Function to get the rotation matrix around the y-axis.

Parameters:

- θ: Angle around the y-axis. Works for rad and deg. Calculater settings are crucial.

Returns:

- rotation matrix (4x4).

---

robotic/rotz(θ)

Function to get the rotation matrix around the z-axis.

Parameters:

- θ: Angle around the z-axis. Works for rad and deg. Calculater settings are crucial.

Returns:

- rotation matrix (4x4).

---

robotic/xyzangles(r)

Function to calculate the retransformation angles according to the X-Y-Z Roll-Gier-Nick

Convention.

Parameters:

- r (3x3): Rotation matrix.

Returns:

- β
- α
- γ

---

robotic/zyzangles(r)

Function to calculate the retransformation angles according to the Z-Y-Z Euler Convention.

Parameters:

- r (3x3): Rotation matrix.

Returns:

- β
- α
- γ

---

robotic/xyzmatrix(α, β, γ)

Function to calculate the retransformation matrix according to the X-Y-Z Roll-Gier-Nick Convention.

Parameters:

- α
- β
- γ

Returns:

- r (3x3): Rotation matrix.

---

robotic/zyxmatrix(α, β, γ)

Function to calculate the retransformation matrix according to the Z-Y-X Euler Convention.

Parameters:

- α
- β
- γ

Returns:

- r (3x3): Rotation matrix.

---

robotic/zyzmatrix(α, β, γ)

Function to calculate the retransformation matrix according to the Z-Y-Z Euler Convention.

Parameters:

- α [rad]
- β [rad]
- γ[rad]

Returns:

- r (3x3): Rotation matrix.

---

robotic/dhttransform(dh)

Function returns all transformation matrices for the intermediate steps and at the end the total transformation matrix.

Parameters:

- dh (nx4): The Denavit-Hartenberg matrix is entered according to the following convention:

Gelenk Nr.	Linklänge a_i	Linkdrehung α_i	Link Offset d_i	Gelenkwinkel θ_i
i				
i+1				
...				

Returns:

- transformation matrices

---

robotic/trapezbahn(s1, s2, v1, v2, a1, a2)

Function returns parameters for a fully synchronous PTP motion with trapezoidal velocity profile.

Parameters:

- s1: distance 1
- s2: distance 2
- v1: speed 1
- v2: speed 2
- a1: acceleration 1
- a2: acceleration 2

Returns:

- acceleration time
- constant travel time
- total time
- synchronized acceleration
- synchronized speed

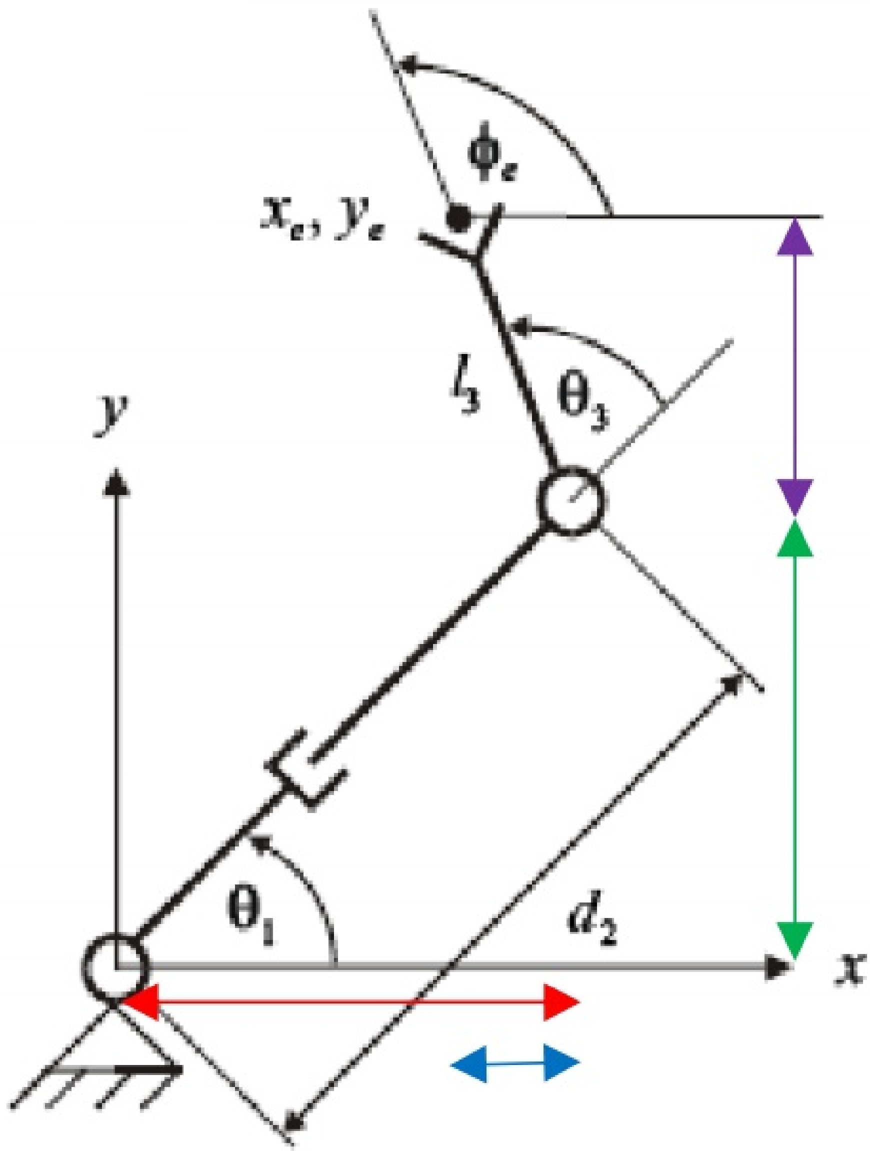
---

robotic/jacobi(xe,ye,Φe,θ1,θ3,d2)

Function to calculate the Jacobi matrix. The Jacobi matrix of a robot arm describes the mapping of joint velocities to the linear velocity of the TCP and the temporal changes of the orientation of the end-effector.

Parameters: [rad]

- xe: Position in X-direction of the TCP.
- ye: Position in Y-direction of the TCP.
- Φe: Angle of the TCP.
- θ1: Angle of the first section of the robot.
- θ3: Angle of the last section of the robot.
- d2: Lenght of the first section of the robot.



Returns:

- Jacobi matrix