

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. Calculator 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. Calculator 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. Calculator 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. Calculator 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 [rad]
 - α [rad]
 - γ [rad]
-

robotic/zyzangles(r)

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

Parameters:

- r (3x3): Rotation matrix.

Returns:

- β [rad]
 - α [rad]
 - γ [rad]
-

robotic/xyzmatrix(α , β , γ)

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

Parameters:

- α [rad]
- β [rad]
- γ [rad]

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 [rad]
- β [rad]
- γ [rad]

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:

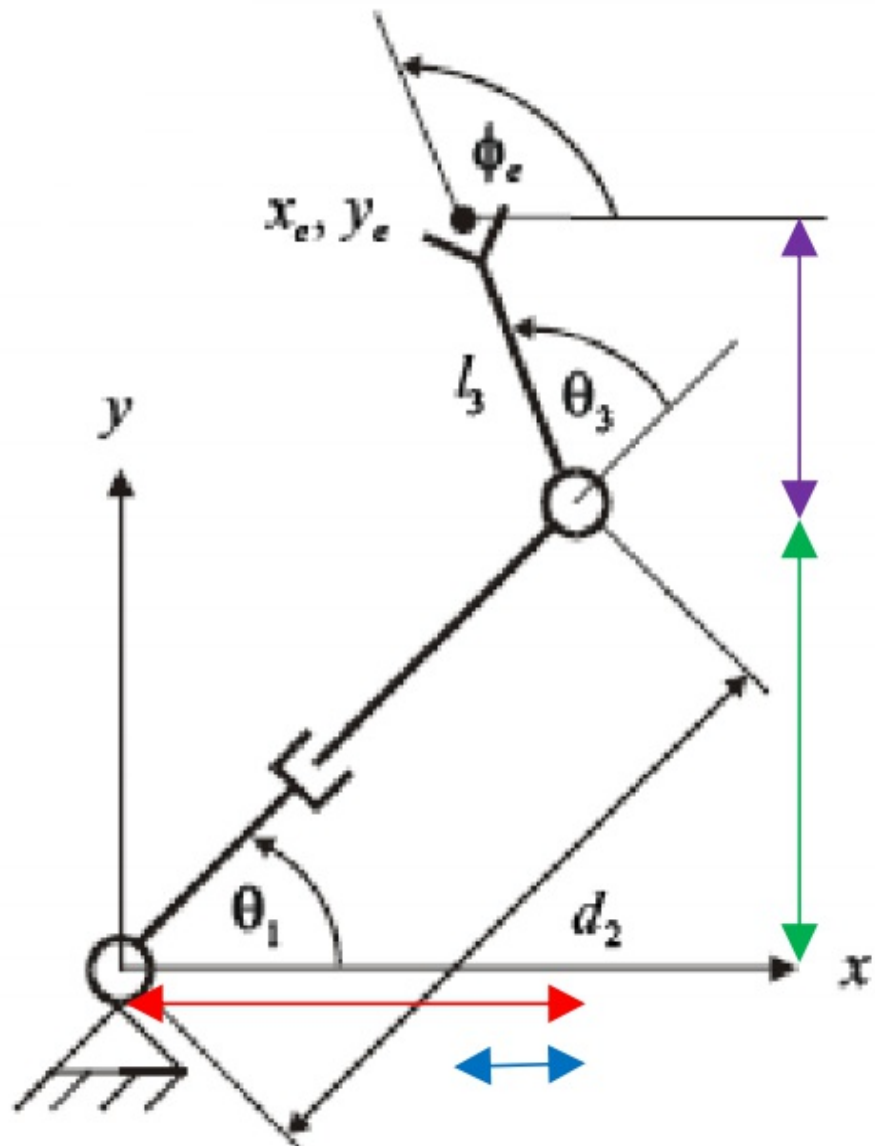
- r (3x3): Rotation matrix.
-

robotic/jacobi($x_e, y_e, \Phi_e, \theta_1, \theta_3, d_2$)

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:

- x_e : Position in X-direction of the TCP.
- y_e : Position in Y-direction of the TCP.
- ϕ_e : Angle of the TCP.
- θ_1 : Angle of the first section of the robot.
- θ_3 : Angle of the last section of the robot.
- d_2 : Length of the first section of the robot.



Returns:

- Jacobi matrix