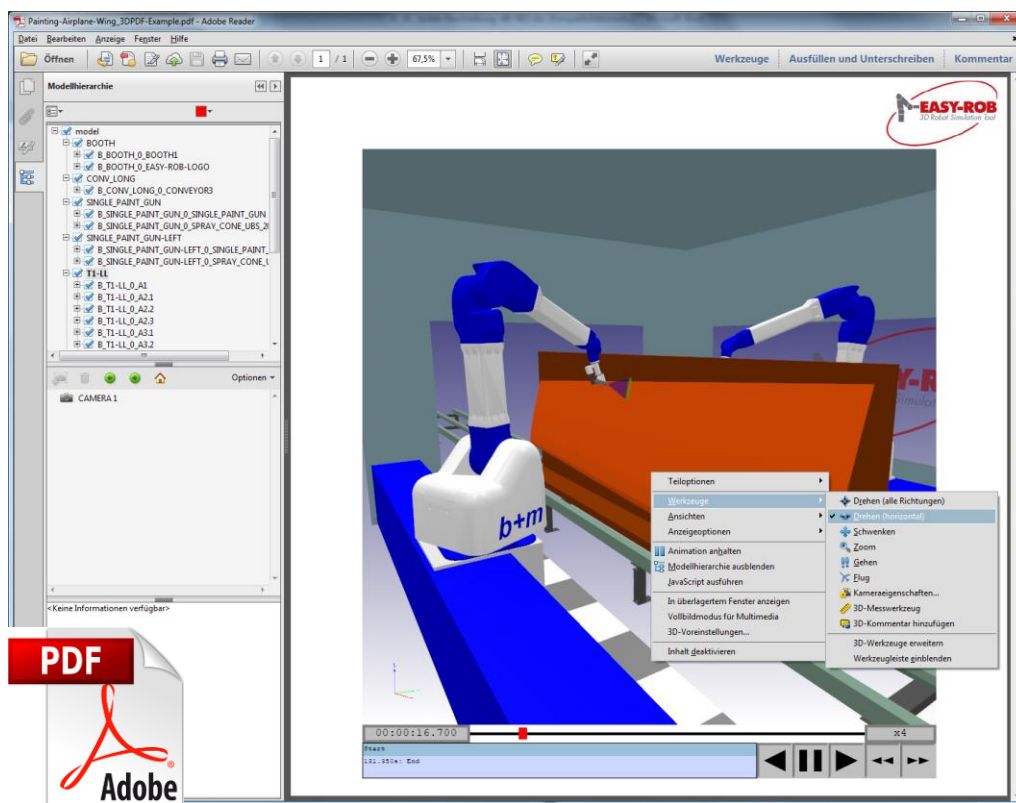


Die neue Version

EASY-ROB™ V6.6



Juni 2015

Version 1.4

EASY-ROB™

Inhaltsverzeichnis

EASY-ROB™ V6.6.....	4
Einstellbare Transparenz der EASY-ROB™-Dialoge	5
AutoRefresh Tag Attribute content.....	6
Kartesische Raumbegrenzung.....	7
Verschiedene Simulations-Ansichten	8
Externer TCP mit Zirkular-Interpolation	9
Optische Hervorhebung eines ausgewählten Gerätes	10
Bewegungsablauf im 3D-PDF mit Animation speichern	11
CAD2ER mit individueller Anpassung der Tesselierung.....	13
Vollständige Roboterbibliotheken	14
Neue kinematische Typen	14
Neue Formeln für Roboter-Dämpfungen	15
Orientierungs-Notation für Universal Robot.....	17
Erweiterte Post-Prozessoren	18
EASY-ROB™ DII Multi-Program-fähig	19
Neue ERCL-Befehle.....	20
ER_CAPI: Berechnung der Jacobi-Matrix.....	22
ER_CAPI: Material- und Licht-Attribute	23
ER_CAPI: Geometry Load Error	24
ER_CAPI: AutoPath™	24
Robotics Simulation Kernel Multi-Thread fähig.....	25
Neues EASY-ROB™-Icon.....	26
Neue Beispielzellen in der ApplicationLib	26
Kontakt	27
Eigene Notizen.....	28

EASY-ROB™ V6.6

Die neue Version steht für alle Kunden mit einer gültigen Lizenz oder einem Softwarepflegevertrag für EASY-ROB™ V6.6 kostenfrei zur Verfügung. Für Kunden älterer Versionen besteht die Möglichkeit ein Update zu erwerben.

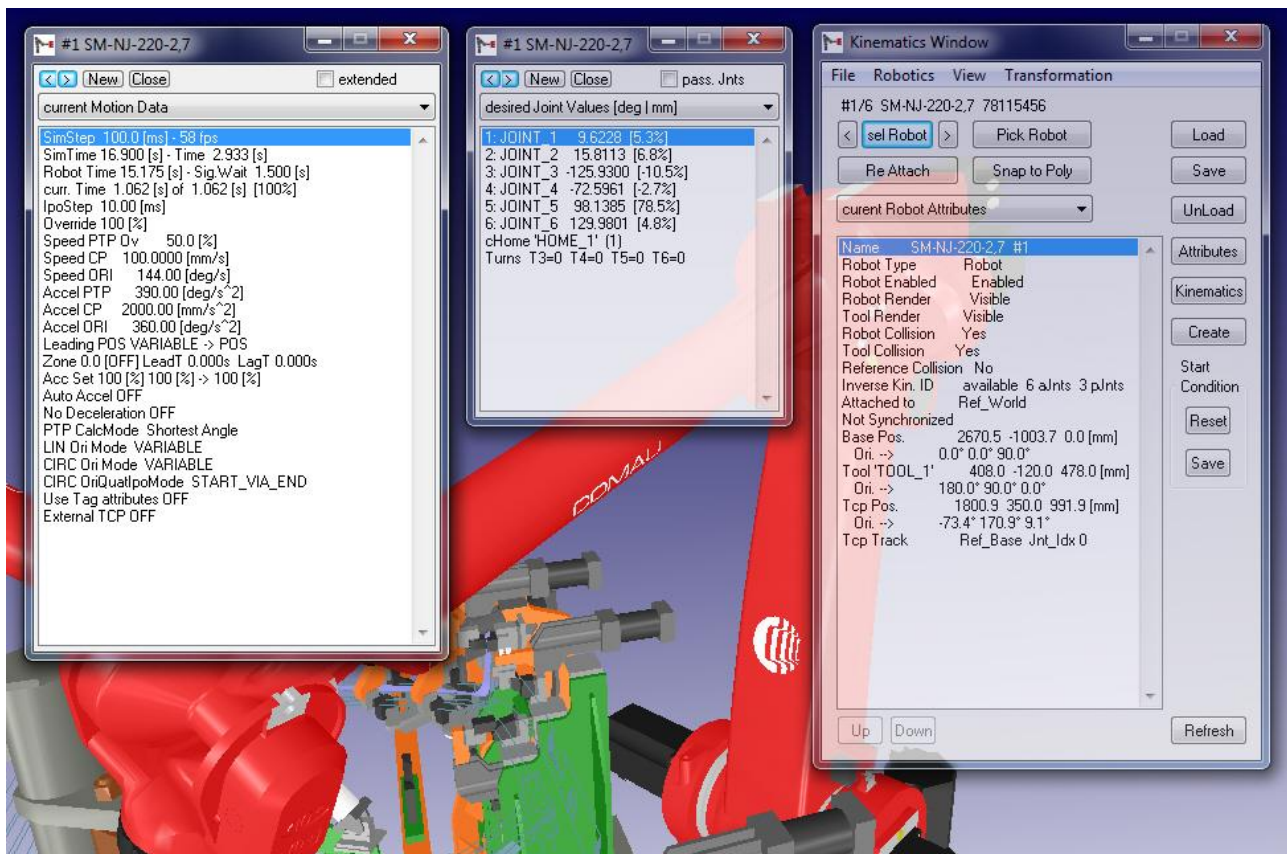
Für Ihre Anregungen und Verbesserungsvorschläge bedanken wir uns schon jetzt bei Ihnen.

Vielen Dank

Stefan Anton
EASY-ROB
3D Robot Simulation Tool

Einstellbare Transparenz der EASY-ROB™-Dialoge

Die Transparenz sämtlicher EASY-ROB™-Dialoge lässt sich nun anschalten und individuell einstellen. Dies trägt erheblich zur Übersichtlichkeit, bei großen Arbeitszellen mit mehreren geöffneten Dialogen bei.



Von links nach rechts: Transparency OFF, Transparency Medium und Transparency High



Dabei lässt sich die Transparenz für jedes einzelne Fenster individuell einstellen, indem Sie auf das EASY-ROB™-Icon des geöffneten Fensters klicken und einen der vordefinierten Transparenz-Werte (**Transparency Low**, **Medium** oder **High**) auswählen.

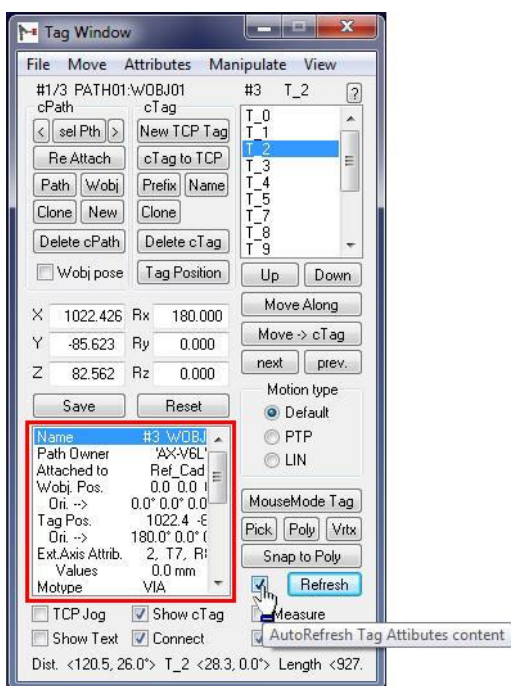
Alternativ kann über **Transparency User** ein eigener Alpha-Wert zwischen 100 (hohe Transparenz) und 255 (keine Transparenz) eingegeben werden.

Die globale Voreinstellung der Transparenz für alle Dialoge finden Sie in der Menüleiste unter **View > Dialog Transparency Settings**. Jeder neu geöffnete Dialog wird nun mit dieser voreingestellten Transparenz geöffnet.

Der individuelle Voreinstellungs-Wert kann in der Environment-Datei easy-rob.env abgespeichert werden.

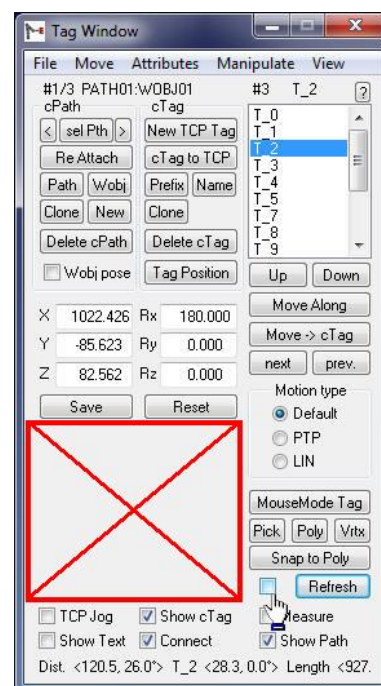
AutoRefresh Tag Attribute content

Ist während einer Simulation das Tag Window geöffnet, so kann es besonders bei sehr vielen Tagpunkten (>1000) zu erheblichen Performance-Einbußen kommen. Dies liegt an der permanenten Aktualisierung und Anzeige der Tag-Attribute im Tag Window. Aus diesem Grund wurde die sogenannte Funktion **AutoRefresh Tag Attributes content** eingeführt, die im Tag Window aktiviert bzw. deaktiviert werden kann.



Ist das Häkchen unter „AutoRefresh Tag Attributes content“ gesetzt, so werden – auch während der Simulation – die Informationen jedes einzelnen Tagpunktes, wie z.B. Tag-Position im Tag Window angezeigt und permanent aktualisiert.

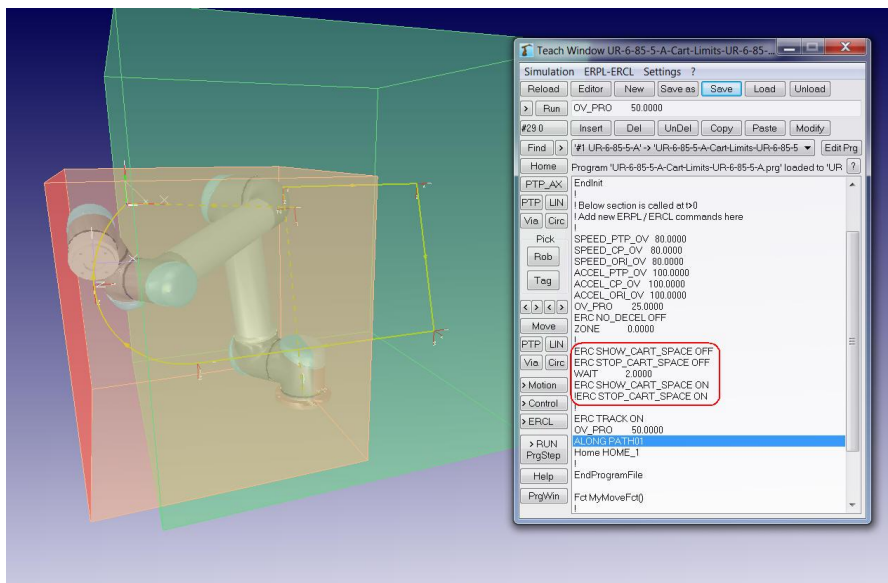
Dies führt besonders bei Arbeitszellen mit sehr vielen Tagpunkten zu erheblichen Performance-Einbußen.



Da die Aktualisierung und Anzeige der Attribut-Informationen jedes Tagpunktes während einer Simulation häufig nicht von Interesse sind, können diese durch Entfernen des Häkchens neben dem Refresh-Button ausgeblendet werden. Dies trägt zur erheblichen Performance-Steigerung bei.

Kartesische Raumbegrenzung

In der neuen Version werden die Cartesian Space Limits eingeführt. Hierbei wird – ähnlich wie bei den Achsverfahrenbereichsgrenzen eines Roboters – ein kartesischer Raum bezüglich der Roboterbasis als Grenze festgelegt, den der Roboter nicht verlassen darf.



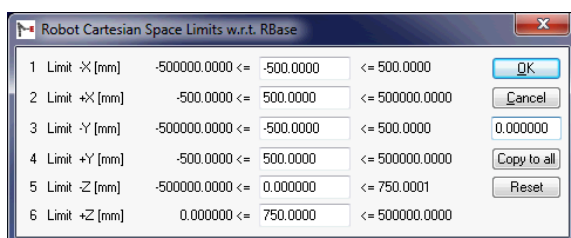
Neue ERCL - Commands:

SHOW_CART_SPACE ON / OFF
STOP_CART_SPACE ON / OFF

CART_LIMITS_MIN X Y Z
CART_LIMITS_MAX X Y Z

Der grüne Raum gibt den vom Benutzer festgelegten Sperrbereich an. Der rote Raum stellt den aktuellen Raum dar, den der Roboter einnimmt. Wird der erlaubte Bereich überschritten, so wird die Fläche an der die Überschreitung stattgefunden hat dunkelrot markiert.

Sie finden die Cartesian Space Limits im Kinematics Window unter Attributes.



Durch Eingabe der Begrenzungswerte [mm] für jede Koordinaten-Richtung wird der kartesische Begrenzungsraum erzeugt.

Die Angabe der Werte erfolgt in Bezug auf die Roboterbasis.

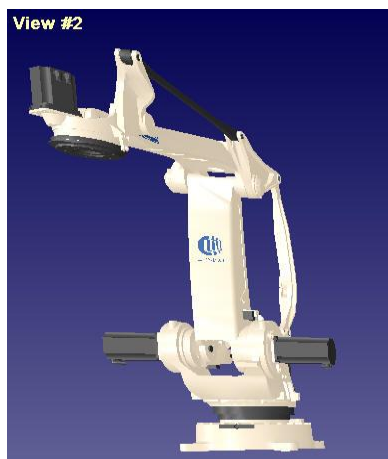
Während der Simulation kann der kartesische Begrenzungsraum an- bzw. ausgeschaltet werden, indem der entsprechende ERCL-Befehl ERC SHOW_CART_SPACE ON / OFF im Teach Window gesetzt wird.

Verschiedene Simulations-Ansichten

In der neuen EASY-ROB™-Version haben Sie die Möglichkeit vier (4) unterschiedliche Simulations-Ansichten für Ihre aktuelle EASY-ROB™-Sitzung einzustellen und diese per Tastatur durchzuschalten.



Simulations-Ansicht #1



Simulations-Ansicht #2



Simulations-Ansicht #3

Die aktuelle Simulations-Ansicht wird durch ein **View #1 ... #4** in der linken oberen Ecke des Bildschirms kurzzeitig angezeigt.

Mit der
Tastenkombination



+



wird zur nächsten Ansicht geschaltet

Mit der
Tastenkombination



+



+



wird zur vorherigen Ansicht geschaltet

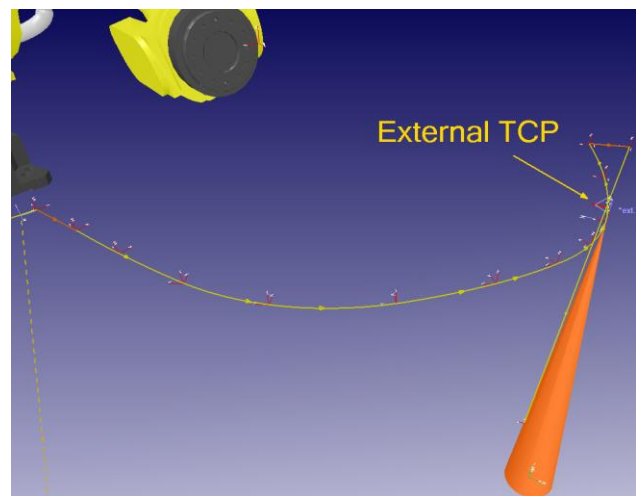
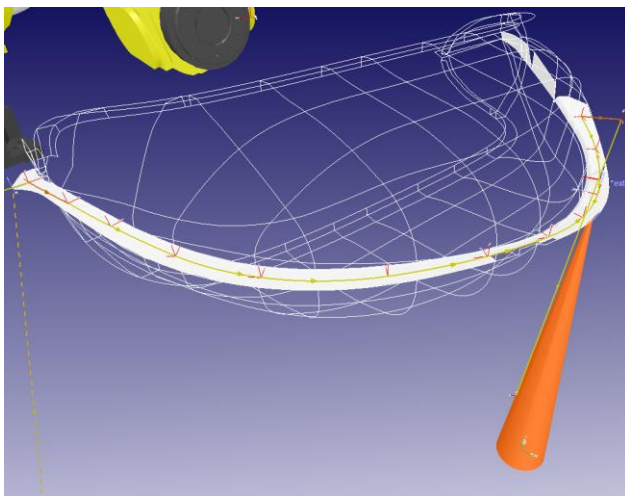
Jede einzelne Simulations-Ansicht kann, wie bisher im Cruise-Mode geändert werden. Die Ansichten können auch während der Simulation geändert werden, um stets die beste Perspektive auf den Prozess zu haben.

Ein Wechsel der Ansichten kann außerdem mit Hilfe die Menüleiste unter
View > Set 3D Views > Goto next Simulation View / Goto prev Simulation View erfolgen.

Externer TCP mit Zirkular-Interpolation

Die werkstückführende Bewegung wurde um die Zirkular-Interpolation erweitert.

Wird für den Roboter ein externer TCP definiert, so kann der Roboter nun das Werkstück auch mittels Zirkularinterpolation entlang des externen TCPs führen.

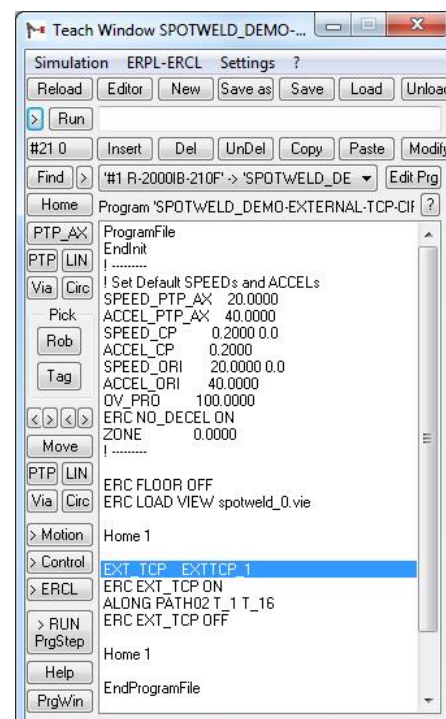


Betrachten Sie hierzu die Beispielzelle „spotweld_demo-external-TCP-CIRC.cel“ im Verzeichnis „\ApplicationLib\SpotWelding“:

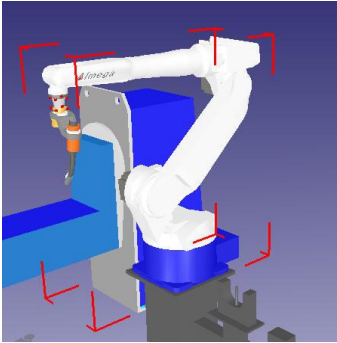
Die Zirkular-Interpolation wird über die Motion-Types VIA und CIRC realisiert. Diese können im Tag Window unter den Tagpunkt-Attributen für die gewünschten Tags festgelegt werden.

Anschließend muss der externe TCP für den Roboter definiert (EXT_TCP ExtTcp_1) und die werkstückführende Bewegung aktiviert (ERC EXT_TCP ON) werden.

Weitere Informationen zu den einzelnen Befehlen zum Thema „Externer TCP“ finden Sie in der Dokumentation zur EASY-ROB™ Program Language „EASY-ROB-ERPL_DE.pdf“.



Optische Hervorhebung eines ausgewählten Gerätes



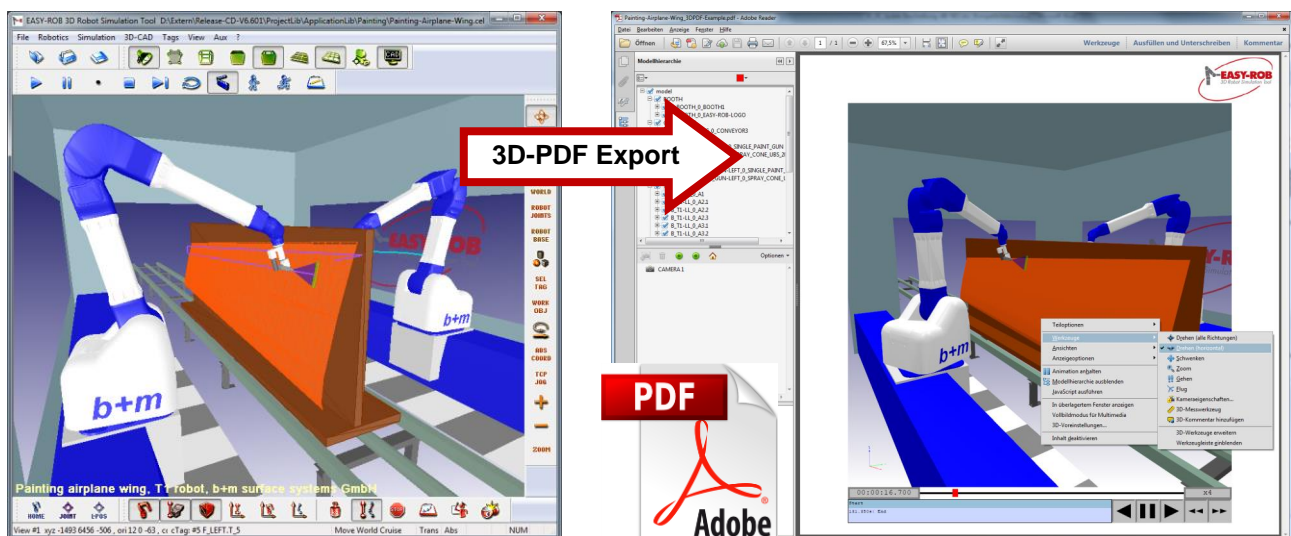
Die Übersichtlichkeit der Arbeitszelle wurde nochmals verbessert.

Zukünftig wird ein aktuell angewähltes Device durch einen roten „Rahmen“ hervorgehoben. So lässt es sich schneller und besser in einer großen Arbeitszelle erkennen.

Tipp: Ein Rechtsmausklick (RMB) im Cruise Mode entfernt den Rahmen

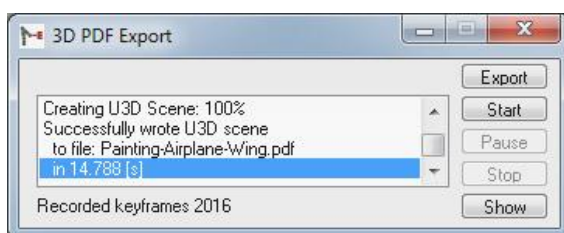
Bewegungsablauf im 3D-PDF mit Animation speichern

Mit der neuen Option EASY-ROB™ **3D-PDF-Export***) kann der gesamte Simulationsablauf mit Animation als PDF-Datei gespeichert und anschließend mit dem kostenfreien Adobe® Reader XI geöffnet und die Simulation gestartet werden.




Wichtig: Für das Abspielen wird mindestens Version XI (11.0.10) des Adobe® Reader empfohlen.

Den 3D-PDF-Export-Dialog finden Sie unter **View > EASY-ROB PlugIn DLLs > 3D-PDF Export**.



3D-PDF Export Dialog

- "Export" erlaubt das Speichern der statischen Szene
- "Start" um die Aufzeichnung zu starten.
- Starten Sie die Simulation . Der Simulationslauf wird nun aufgezeichnet. Die Anzahl der aufgezeichneten Key-Frames wird Ihnen angezeigt.
- "Stop" um die Aufzeichnung zu beenden. Sie werden aufgefordert das Zielverzeichnis und den Namen des PDF-Dokumentes einzugeben.
- Das 3D-PDF-Dokument wird erzeugt.

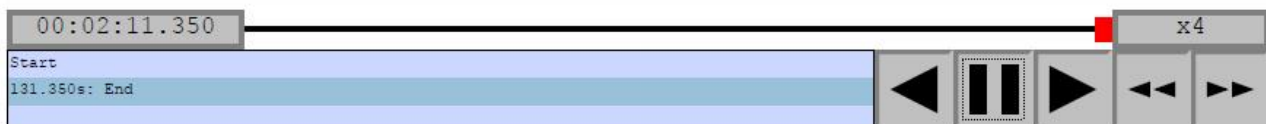
Innerhalb des PDF-Dokuments können Sie die aufgezeichnete Szene beliebig rotieren und zoomen.

Im Verzeichnis „\ApplicationLib\Painting“ finden Sie ein das Beispiel-3D-PDF-Dokument „Painting-Airplane-Wing_3DPDF-Example.pdf“. Dieses wurde aus der Arbeitszelle „Painting-Airplane-Wing.cel“ erzeugt.

*) Das PlugIn 3D-PDF-Export muss kostenpflichtig lizenziert werden

Bewegungsablauf mit Animation in 3D-PDF speichern

Im Adobe® Reader können Sie mit der Navigationsleiste den aufgezeichneten Bewegungsablauf Starten, Pausieren, Stoppen, Vor- und Zurückspulen, sowie die Geschwindigkeit (x1/64-fache bis x64-fache) ändern. Die Zeitangabe gibt die dabei die reale Prozesszeit an.



Navigationsleiste im Adobe® Reader

Die 3D-PDF-Export-Funktionalität kann alternativ auch über ERCL-Kommandos gesteuert werden. Dabei stehen dem Benutzer die folgenden ERCL-Befehle zur Verfügung:

ERC _3D_PDF_EXPORT SCREENSHOT [filename]

ERC _3D_PDF_EXPORT ON / OFF [filename]

ERC _3D_PDF_EXPORT SET_FILE filename

ERC _3D_PDF_EXPORT SET_LABEL labelname

ERC _3D_PDF_EXPORT SET_PASSWORD passwordname

ERC _3D_PDF_EXPORT PAUSE

ERC _3D_PDF_EXPORT DEACTIVATE

Eine Beschreibung der Kommandos finden Sie im Abschnitt "Neue ERCL-Befehle" und im Dokument zur ERPL- / ERCL- Programmiersprache „**EASY-ROB-ERPL_DE.pdf**“.

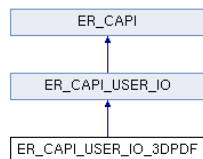
Der 3D-PDF Export lässt sich auch über die API Methodenklasse: ER_CAPI_USER_IO_3DPDF ansteuern.

ER_CAPI_USER_IO_3DPDF Class Reference

Method class for 3D Pdf Export. More...

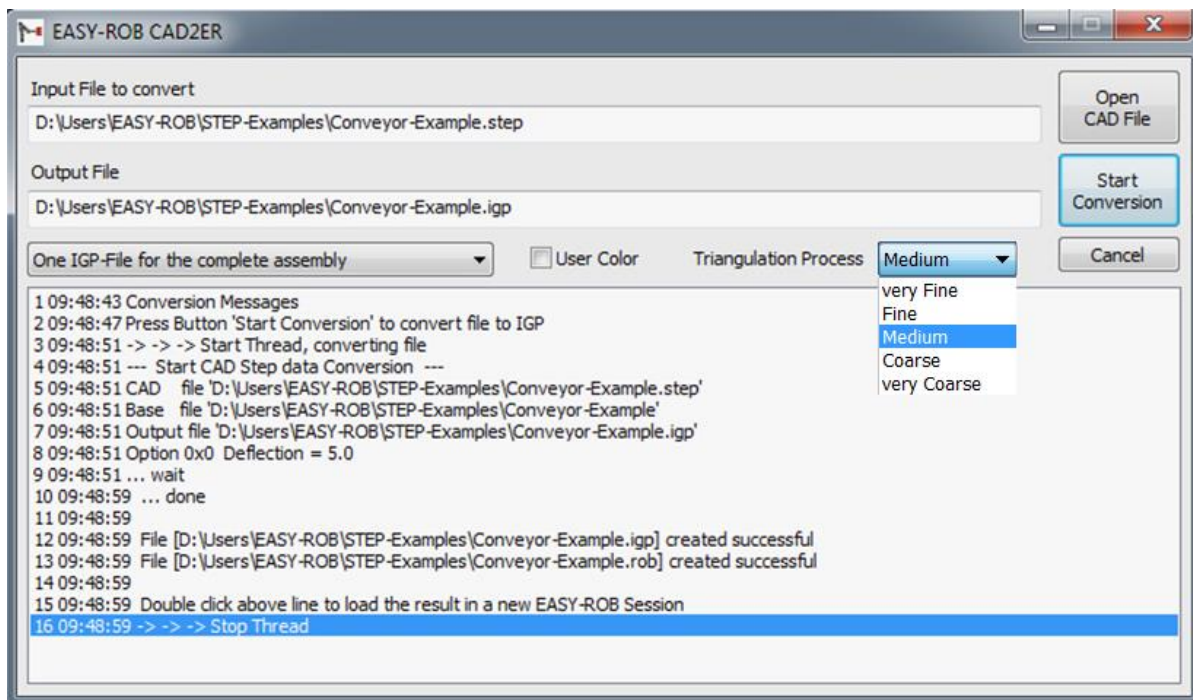
```
#include <ER_CAPI.H>
```

Inheritance diagram for ER_CAPI_USER_IO_3DPDF:



http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/class_er__c_a_p_i__u_s_e_r__i_o__3_d_p_d_f.html

CAD2ER mit individueller Anpassung der Tesselierung



Den neuen CAD2ER Konverter gibt es auch als 64-Bit Version, was besonders bei großen Dateien vorteilhaft ist. EASY-ROB™ CAD2ER erlaubt es die neutralen **STEP- (AP203, AP214)**, **IGES-** und **VRMLII,97**-Datenformate zu importieren und nach EASY-ROB™ **IGP** zu konvertieren.

Neu ist außerdem die individuelle Anpassung der importierten CAD-Geometrien. Hierbei kann die Tesselierung nach Belieben beeinflusst werden, was insbesondere bei sehr detaillierten und „großen“ Geometrien hilfreich ist um angemessene Simulationsgeschwindigkeiten zu erzielen.

Unter **Triangulation Process** kann zwischen den Detaillierungsgraden **very fine**, **Fine**, **Medium**, **Coarse** und **very Coarse** unterschieden werden.



Coarse – 23.337 Polygone



Normal – 35.692 Polygone



Fine – 67.470 Polygone

Beim Import von Baugruppen oder Assemblies werden sofort Device-Files erzeugt, so dass sich die Komponenten individuell ausblenden oder gar löschen lassen.

Vollständige Roboterbibliotheken

In EASY-ROB™ stehen vollständige Bibliotheken zur Einbindung aller bedeutenden Robotertypen des Marktes bereit. Dazu zählen ABB, b+m, Comau, Denso, Eisenmann, Fanuc, Guedel, igm, Kawasaki, KUKA, Mitsubishi, Motoman, OTC-Daihen, Reis, Stäubli, Tricept, Unimation und Universal Robots.

Nicht vorhandene Roboter, Handlingsysteme, Maschinen und Werkzeuge lassen sich in EASY-ROB™ einfach und schnell „virtuell nachbauen“.

Die Roboterbibliotheken von ABB, KUKA, Comau, Fanuc, Motoman und Stäubli sind nahezu vollständig und werden permanent von uns gepflegt.

Derzeit sind mehr als 900 Roboter, Positionierer und externe Tracking-Achsen verschiedenster Hersteller verfügbar.

Weitere Informationen:

<http://www.easy-rob.com/produkt/erweiterungen/roboter-bibliotheken.html>

Neue kinematische Typen

Die neuen Modelle der KUKA LBR iiwa-Serie und Palettierer der Comau Smart5 PAL-Reihe gesellen sich zu den Robotertypen hinzu.



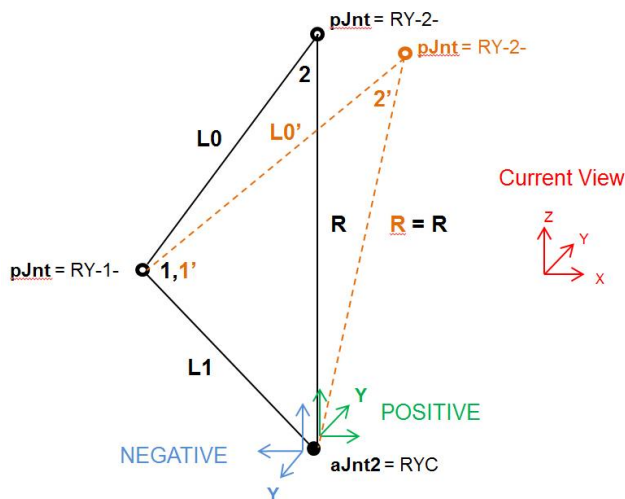
KUKA LBR iiwa



Comau Smart5 PAL

Neue Formeln für Roboter-Dämpfungen

Zukünftig wird die kinematische Modellierung von Robotern mit Dämpfungen und Ausgleichsmassen wie zum Beispiel dem ABB-IRB-7600-500-255.rob vereinfacht. Dafür wurde eine Formel entwickelt, die - entsprechend dem kinematischen Modell - mit Parametern ergänzt werden muss und anschließend die Winkeländerungen der relevanten passiven Achsen in Abhängigkeit der aktiven zweiten Achse (aJnt2) berechnet. So lässt sich eine korrekte Bewegung der passiven Achsen, wie sie z.B. für Dämpfungen benutzt werden realisieren und darstellen.



Um die Parameter zu bestimmen, muss die Nullstellung- bzw. Ausgangsposition des Roboters und das sich daraus ergebende kinematische Drahtmodell betrachtet werden.

Die JNTDAMP_TRI - Funktion lautet:

pJnt_i = JNTDAMP_TRI
(L0, L1, R, position, aJ2_ori, pjnt_idx)

Folgende Parameter müssen festgelegt und in die JNTDAMP_TRI-Funktion eingegeben werden:

L0 = Abstand pJnt RY-1- zu pJnt RY-2- [mm]
L1 = Abstand aJnt2 RYC zu pJnt RY-1- [mm]
R = Abstand aJnt2 zu RYC pJnt RY-2- [mm]

position - Lage des pJnt RY-2- zu aJnt2 RYC
 Werte: [DAMP_BELOW, DAMP_ABOVE] mit
 DAMP_BELOW = 0: unter aJnt2
 DAMP_ABOVE = 1: über aJnt2

aJ2_ori - Richtung der y-Achse (Drehachse) des aJnt2 RYC bei Betrachtung der Ausgangslage (Current View) Werte: [POSITIVE, NEGATIVE] mit
 POSITIVE = 1: in pos. Richtung
 NEGATIVE = -1: in neg. Richtung

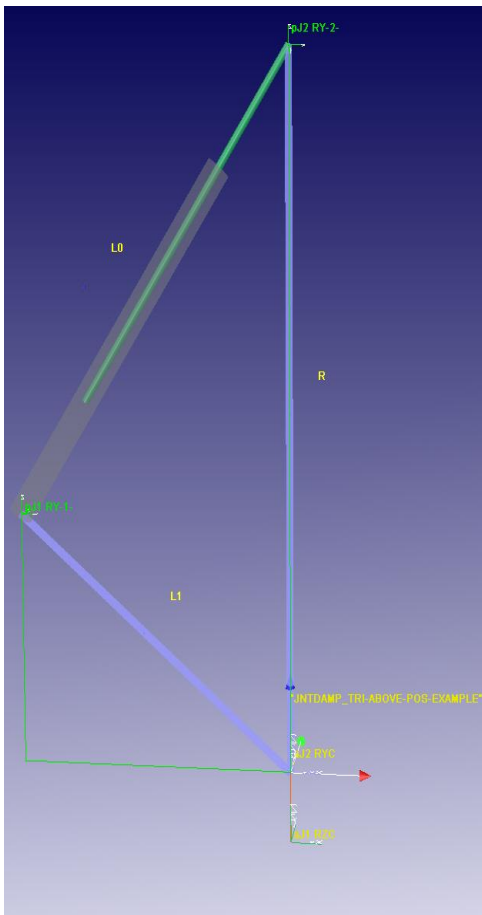
pjnt_idx - Index/Nummer der passiven Achse für Ergebnis (Return)-Wert
 Werte: [1..2] mit

- 1 = Funktion gibt Winkeländerung für den Winkel zwischen L0 und L1 aus
- 2 = Funktion gibt Winkeländerung für den Winkel zwischen L0 und R aus

Die zur jeweiligen passiven Achse gehörige Formel wird mit den entsprechenden Parametern im Kinematics Window unter **Kinematics > 2 – Passive Joints > passive Joint () > 5 – Math. Jnt dependency** eingetragen.

Neue Formeln für Roboter-Dämpfungen

Ein Beispiel zur Funktion „JNTDAMP_TRI“:



JNTDAMP_TRI-Funktion Beispiel

$pJnt_i = JntDampTri (L0, L1, R, position, aJnt_ori, pJnt_idx)$

Zunächst werden alle relevanten Längen in EASY-ROB ausgemessen:

L0 = 400.000 mm, **L1** = 300.000 mm, **R** = 500.000 mm

Anschließend muss die Lage der passiven Achse pJ2 RY-2- gegenüber aJnt2 bestimmt werden:

pJ2 RY-2- liegt oberhalb aJ2 RYC, damit ist

position = DAMP_ABOVE

Bei Betrachtung des aJ2-Koordinatensystems sehen wir, dass die y-Achse vom Betrachter weg (also in positive Richtung) zeigt.

aJnt_ori = POSITIVE

Die nun gegebenen Parameter fügen wir in die JNTDAMP_TRI-Funktion ein und setzen diese im Kinematics Window unter **Kinematics > 2 – Passive Joints > passive Joint () > 5 – Math. Jnt dependency** für die jeweilige passive Achse ein:

Für pJ1 RY-1-:

=L0=400;R=500;L1=300;
JNTDAMP_TRI(L0,L1,R,DAMP_ABOVE, POSITIVE,1)

Für pJ2 RY-2-:

= **JNTDAMP_TRI(L0,L1,R,DAMP_ABOVE,POSITIVE,2)**

Wird nun aJ2 bewegt, ändern sich pJ1 und pJ2 mit den aus der JNTDAMP_TRI-Funktion berechneten Winkeländerungen:

Beispiel:

Die Achse aJ2 wird um +10° bewegt, also $dof(2)*DEG=10.000^\circ$

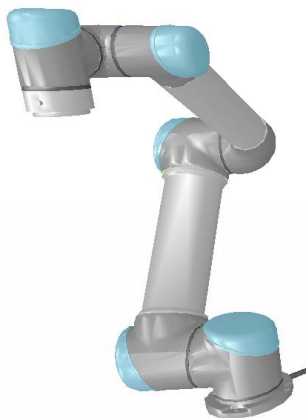
Die JNTDAMP_TRI-Funktion berechnet aus den gegebenen Parametern folgenden Werte:

pJ1 = 9.42° und pJ2 = - 0.58°

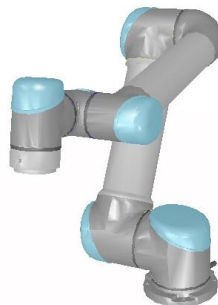
Schauen Sie sich zu diesem Thema die neuen Beispiele „JNTDAMP_TRI-ABOVE-NEG-Example.rob“, „JNTDAMP_TRI-ABOVE-POS-Example.rob“, „JNTDAMP_TRI-BELOW-NEG-Example.rob“ und „JNTDAMP_TRI-BELOW-POS-Example.rob“ im Verzeichnis „Tutorial/Functional“ an.

Orientierungs-Notation für Universal Robot

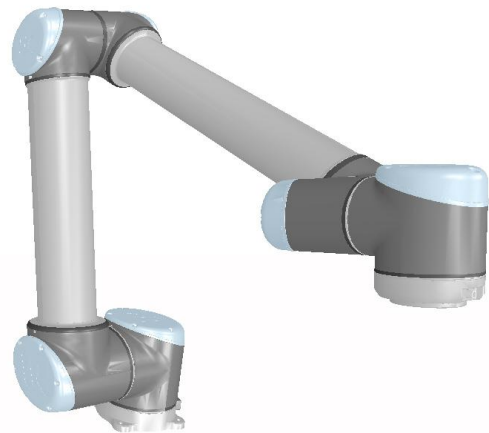
Die Orientierungs-Notation "One axis angle" für Roboter des Herstellers Universal Robot wurde hinzugefügt.



UR-5

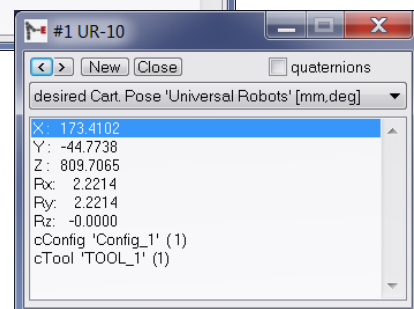
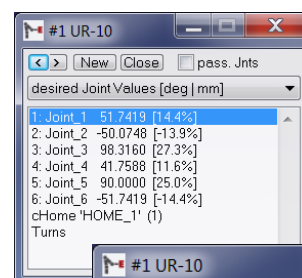
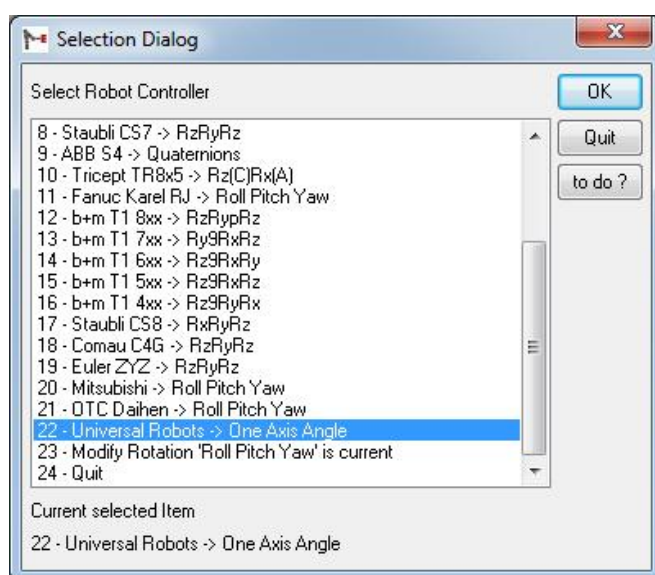


UR-3



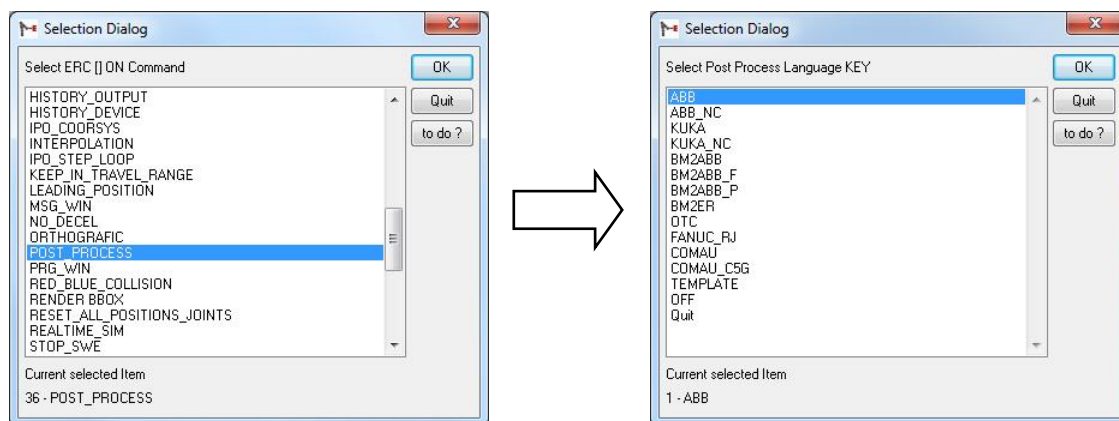
UR-10

Wählen Sie dazu unter **Robotics > cRobot Motion Planner > Controller** die Notation **22 – Universal Robots -> One Axis Angle** aus:

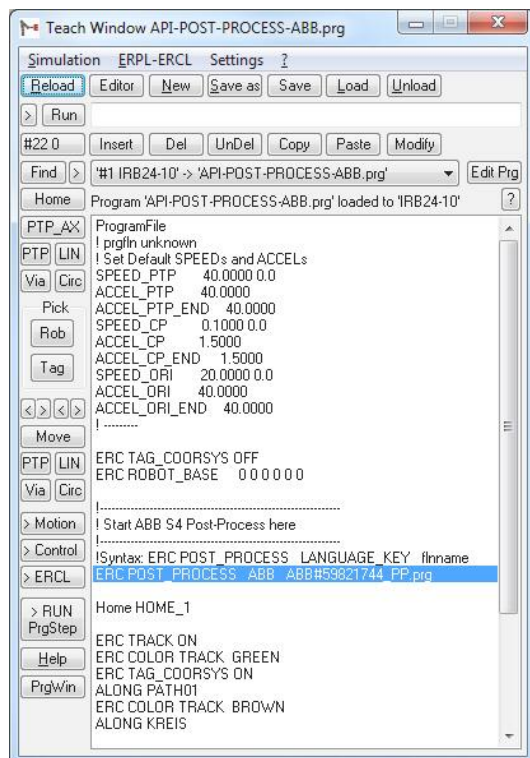


Erweiterte Post-Prozessoren

Beim Setzen des ERCL-Befehls ERC POST_PROCESS im Teach Window wird nun ein Selection Dialog geöffnet, aus dem direkt die gewünschte Post Process Language (ABB, KUKA, etc.) ausgewählt werden kann:



Im Teach Window wird automatisch die entsprechende Befehlszeile (z.B. für den ABB-Post-Prozessor) eingefügt:



Weitere Informationen zur Syntax finden Sie in der ERPL-Dokumentation „EASY-ROB-ERPL_DE.pdf“.

Derzeit sind Post Prozessor-Beispiele/Templates für verfügbar die Roboter der Hersteller:

- ABB
- KUKA
- BM
- OTC
- Fanuc
- Comau

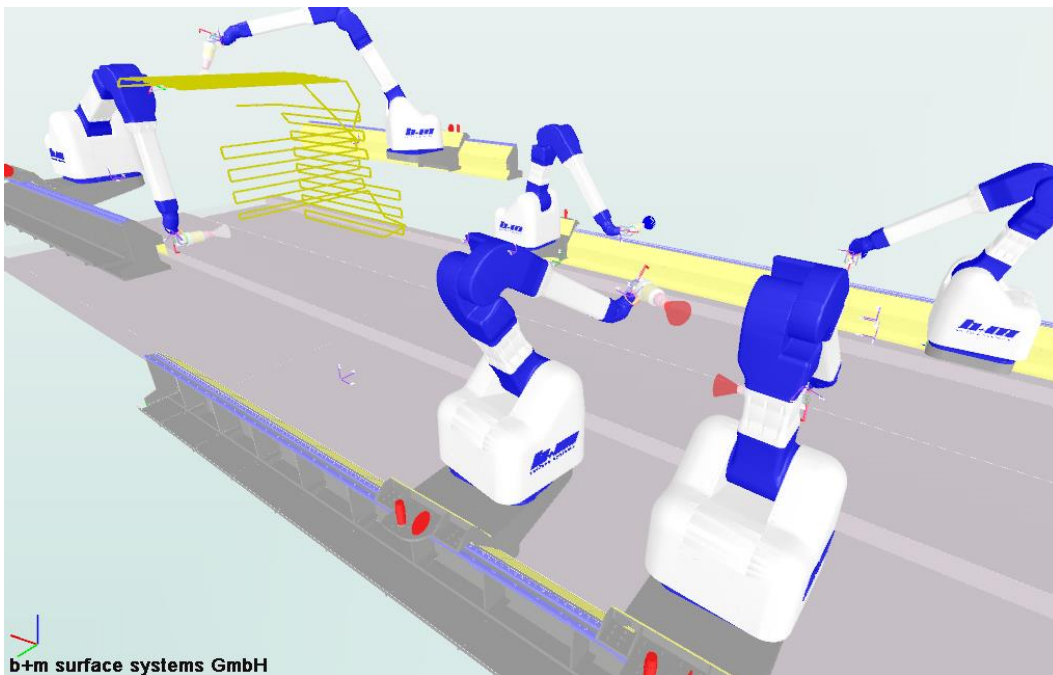
Weitere Post Prozessoren können vom Bediener über die API-Post-Process individuell angepasst und hinzugefügt werden.

EASY-ROB™ DII Multi-Program-fähig

Die EASY-ROB™ DLL Version wurde um die Multi-Program-Funktionalität erweitert.

Wie bei der EXE-Version, ist nun auch bei der DLL-Version die Anzahl der Roboter pro Arbeitszelle nicht limitiert und jedes Gerät kann ein eigenes ERPL-Programm geladen haben (Multi-Program). Kinematiken wie Greifer, Positionierer und/oder Zuführeinheiten können beliebig oft geladen, verkettet und simuliert werden. Diese lassen sich synchronisiert und gleichzeitig bewegen, wobei die Programme über I/O-Signale miteinander kommunizieren.

Integrationsbeispiel einer Multi-Program-fähigen EASY-ROB™ DLL Version:



b+m surface systems GmbH

EasyEdit™ Fahrerhauslackierung, b+m surface systems GmbH

Neue ERCL-Befehle

3D-PDF Export

ERC _3D_PDF_EXPORT SCREENSHOT [flnname]

Erzeugt ein 3D-PDF-Dokument der aktuellen Szene (ohne Animation), d.h. ein Standbild mit allen Achswinkelstellungen zum Zeitpunkt des Befehls. Wird kein [flnname] eingegeben, wird der Name der Arbeitszelle für das 3D-PDF-Dokument übernommen.

ERC _3D_PDF_EXPORT ON / OFF [flnname]

Aktivieren/ Deaktivieren der Aufzeichnung des 3D-PDF-Dokumentes. Wird kein [flnname] eingegeben, wird der Name der Arbeitszelle für das 3D-PDF-Dokument übernommen. Beim Beenden der Aufzeichnung (OFF) wird das 3D-PDF-Dokument automatisch erzeugt und unter dem angegebenen Namen gespeichert.

Wichtig: Wurde zuvor ein 3D-PDF-Dokument erzeugt und ist im Adobe® Reader geöffnet, so muss es geschlossen werden, damit die Aufzeichnung beendet und das Dokument erneut gespeichert werden kann.

ERC _3D_PDF_EXPORT SET_FILE flnname

Das 3D-PDF-Dokument wird mit dem unter „flnname“ angegebenen Namen erzeugt. Zuvor muss der Befehl _3D_PDF_EXPORT ON gesetzt werden.

ERC _3D_PDF_EXPORT SET_LABEL labelname

Setzt ein Label mit dem unter „labelname“ angegebenen Text. Der Text des Labels wird im geöffneten 3D-PDF-Dokument neben der realen Prozesszeit angezeigt und dient der individuellen Kennzeichnung einzelner Prozess-Abschnitte. Zuvor muss der Befehl _3D_PDF_EXPORT ON gesetzt werden.

ERC _3D_PDF_EXPORT SET_PASSWORD passwordname

Das 3D-PDF-Dokument wird mit dem unter „passwordname“ angegebenen Passwort geschützt. Dieses muss beim Öffnen des Dokuments eingegeben werden. Zuvor muss der Befehl _3D_PDF_EXPORT ON gesetzt werden.

ERC _3D_PDF_EXPORT PAUSE

Pausiert die Aufzeichnung des 3D-PDF-Dokuments. Ein erneutes Setzen des Befehls setzt die Aufnahme fort. Zuvor muss der Befehl _3D_PDF_EXPORT ON gesetzt werden.

ERC _3D_PDF_EXPORT DEACTIVATE

Die Aufnahme des 3D-PDF-Dokumentes wird abgebrochen und das Dokument verworfen.

Cartesian Space Limits

Hinweis:

Eine Beschreibung zur kartesischen Raumbegrenzung finden Sie auf Seite 7 dieser Update-Beschreibung.

ERC SHOW_CART_SPACE ON/OFF

Ein-/ Ausschalten der Visualisierung der kartesischen Raumbegrenzung, die auch im Kinematics Window unter Attributes definiert werden kann. Der Sperrbereich wird in der Simulation als grüner Raum dargestellt.

ERC STOP_CART_SPACE ON/OFF

Aktiviert/ Deaktiviert den automatischen Stopp bei Überschreiten des kartesischen Begrenzungsraums.

ERC CART_LIMITS MIN

Legt die minimalen kartesischen Begrenzungswerte [m] für die XYZ Koordinaten-Richtung fest. Durch zusätzliche Angabe der maximalen Begrenzungswerte (ERC CART_LIMITS MAX), wird der kartesische Begrenzungsraum erzeugt.

ERC CART_LIMITS MAX

Legt die maximalen kartesischen Begrenzungswerte [m] für die XYZ Koordinaten-Richtung fest. Sind die minimalen Begrenzungswerte angegeben, wird der kartesische Begrenzungsraum erzeugt.

ER_CAPI: Berechnung der Jacobi-Matrix

```
int ER_CAPI_ROB_KIN::rob_kin_jacobian_q ( )
```

Diese Methode berechnet - abhängig von der Achsposition q - die Werte der Jacobi-Matrix im Flansch (Tip) oder im TCP bzgl. Roboter-Basis.

```
static ER_DllExport int ER_CAPI_ROB_KIN::rob_kin_jacobian_q ( float *   qn,
                                                             frame *   tTw,
                                                             float *   jac,
                                                             int       n_dofs,
                                                             float     delta_scale = 1
                                                             )
```

Calculation of Jacobian matrix

This method calculates, depending on robots joint location q , the jacobian matrix in the flange (Tip) or in Tcp w.r.t. the robots base. If parameter tTw is NULL, the jacobian is in the Tip.

Remarks

For known kinematics, the jacobian is calculated analytically.

For unknown or user defined kinematics, those with mathematically joint dependencies for example, the jacobian matrix is calculated computerized.

In this case a deviation of $dR=0.01$ deg for rotational and $dx=1$ mm for translational joints will be taken.

Parameter $delta_scale$ allows to change these values.

```
// Example:
float *q      = er_rob_kin.inq_q_solut();           // pointer to desired joint data
int num_dofs  = er_rob_kin.inq_num_active_jnts();   // number of robot joints
frame *tTw    = er_rob_kin.inq_tTw();              // Current Tool frame, Robot Tip (flange) to TCP

float *Jac    = new float[DOF6*num_dofs];         // Jacobian (DOF6 x num_dofs)
if (Jac)
{
    float delta_scale = 0.5f; // scaling for computerized calculation of jacobian matrix
    er_rob_kin.rob_kin_jacobian_q(q,tTw,Jac,num_dofs,delta_scale); // in Tcp
    er_user_io_dialog._info_line_msg_M(0,"Jac in Tcp",Jac,DOF6,num_dofs);

    er_rob_kin.rob_kin_jacobian_q(q,NULL,Jac,num_dofs,delta_scale); // in Tip
    er_user_io_dialog._info_line_msg_M(0,"Jac in Tip",Jac,DOF6,num_dofs);
    delete Jac;
}
...
```

Parameters

[in] qn	input joint values
[in] tTw	tool, from Robot Tip (flange) to Tcp
[out] jac	Jacobian matrix in Tip or Tcp w.r.t. Robot base
[in] n_dofs	number dof
[in] delta_scale	scaling for computerized calculation of jacobian matrix

Return values

0 - OK
-1 - Error

Weitere Informationen:

http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/class_e_r__c_a_p_i__r_o_b__k_i_n.html

ER_CAPI: Material- und Licht-Attribute

float* ER_CAPI_SYS_VIEW::inq_ergl_material_light ()

Request OpenGL material or light attributes

```
static ER_DllExport float* ER_CAPI_SYS_VIEW::inq_ergl_material_light ( int ergl_param = ERGL_UNDEF,
                                                                    int ergl_mode = ERGL_MODE_INQ
                                                                    )
```

static

Request OpenGL material or light attributes

Remarks

Parameter **ergl_param** defines the requested value

Use parameter **ergl_mode** to get the address **ERGL_MODE_INQ** or to reset **ERGL_MODE_RESET** or to transfer the attributes to OpenGL **ERGL_MODE_SET**.

```
// Example:
// Get specular RGB componetes for light source #1
float *p = er_sys_view.inq_ergl_material_light(ERGL_LIGHT_1_SPECULAR, ERGL_MODE_INQ);
// change the RED component within 0 and 1
float *v = &p[0];
*v += 0.01;
if (*v<0) v*=0;
else if (*v>1) v*=1;
// change some more attributes here .... or reset to original attributes
// er_sys_view.inq_ergl_material_light(ERGL_LIGHT_1_SPECULAR, ERGL_MODE_RESET);
// Finally, transfer modified attributes to OpenGL
er_sys_view.inq_ergl_material_light(ERGL_ALL_ATTRIBUTES, ERGL_MODE_SET);
// make a graphical update to see the results
er_sys_view.grf_update_export();
...
```

Parameters

[in] **ergl_param** is one of **ERGL_MATERIAL_AMBIENT** :: **ERGL_MATERIAL_DIFFUSE** to **ERGL_LIGHT_4_EMISSION**

[in] **ergl_mode** is one of **ERGL_MODE_INQ**, **ERGL_MODE_RESET**, **ERGL_MODE_SET**

Return values

pointer to requested value

Weitere Informationen:

http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/class_e_r__c_a_p_i__s_y_s__v_i_e_w.html

ER_CAPI: Geometry Load Error

int `AUX_UPDATE_IDX_LOAD_GEO_ERROR`

Wird beim Laden von Geräten eine Geometrie nicht gefunden, so wird entsprechendes Feedback ausgegeben.

```
const int AUX_UPDATE_IDX_LOAD_GEO_ERROR = 54
```

Geometry file (IGP, STL or 3DS) not found during load.

Weitere Informationen:

http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/index.html

ER_CAPI: AutoPath™

Methoden-Klasse zur Ansteuerung der kollisionsfreien Pfadplanung.

Folgende Funktionen aus ER_CAPI_MOP_AUTOPATH Class wurden nach er_dvlp_ext.h kopiert:

```
EXPORT_C char* AutoPathVer(void);
EXPORT_C void* inq_AutoPathDlg_Hwnd(void);
EXPORT_C int SetDlgParameter(int ap_dlg_option, int ap_dlg_value);
EXPORT_C int GetDlgParameter(int ap_dlg_option);
EXPORT_C float* SetStartPose(void);
EXPORT_C float* SetEndPose(void);
EXPORT_C int FindPath(void);
EXPORT_C int AbortPlanning(void);
EXPORT_C int GetPlanningStatus(void);
EXPORT_C int GetNumberOfWayPoints(void);
EXPORT_C int GetWayPointDof(void);
EXPORT_C float* GetWayPoint(int idx);
EXPORT_C int ClearAllWayPoints(void);
EXPORT_C UINT* Accuracy(void);
EXPORT_C int SetAxisPriority(int axisBit, int priority);
EXPORT_C int SetAxisEnable(int axisBit, int enable);
EXPORT_C int SetParameter(int ap_option, int ap_value);
EXPORT_C int GetParameter(int ap_option);
```

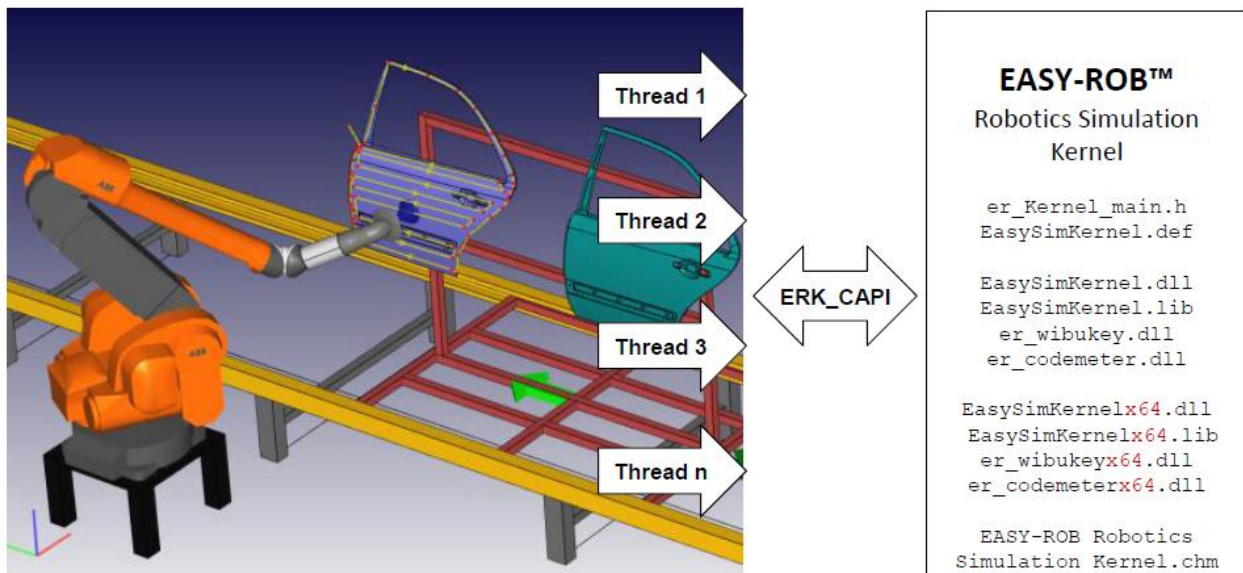
Weitere Informationen zu den einzelnen Funktionen:

http://www.easy-rob.com/fileadmin/Userfiles/doc/er_capi/class_er_capi_mop_autopath.html#details

Robotics Simulation Kernel Multi-Thread fähig

Mit der neuen Version ist der EASY-ROB Robotics Simulation Kernel (ERK) nun Multi-Thread fähig.

Beim Aufruf von Funktionen aus der ERK DLL durch benutzerspezifische Programme, muss sichergestellt werden, dass ein gleichzeitiges und mehrfaches Ausführen der Funktionen möglich ist, ohne dass diese sich gegenseitig behindern. Beim sogenannten Multithreading werden die vom Benutzer erzeugten Threads, also die einzelnen Ausführungsstränge eines Prozesses parallel ausgeführt und bearbeitet. Greifen zwei Threads auf einen gemeinsamen Speicherbereich zu, so werden deren Änderungen koordiniert um Fehler zu vermeiden.

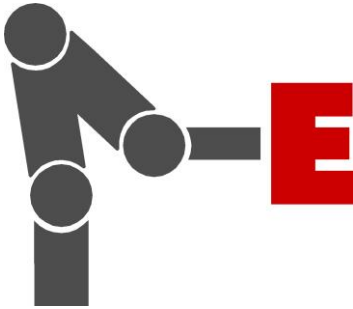


OEM Applikation: technologiebasierte
Software mit eigener 3D Visualisierung
Parallele Ausführung und Bearbeitung von
Threads möglich

Auszug aus der MSDN-Library zum Thema Threadsicherheit in der C++-Standardbibliothek:

„Wenn durch einen Thread in ein Objekt geschrieben wird, müssen alle Lese- und Schreibvorgänge im Zusammenhang mit diesem Objekt in diesem oder anderen Threads geschützt werden. Wenn z. B. Thread 1 in Objekt A schreibt, muss verhindert werden, dass Thread 2 Objekt A ausliest oder in Objekt A schreibt.“

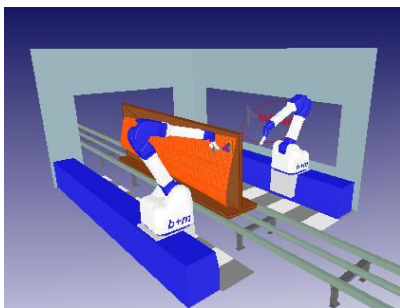
Neues EASY-ROB™-Icon



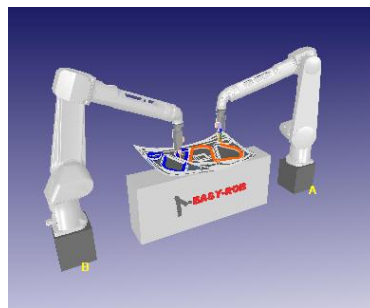
Mit Release der neuen EASY-ROB™-Version 6.6 wird dieses Icon Ihren Desktop schmücken.

Neue Beispielzellen in der ApplicationLib

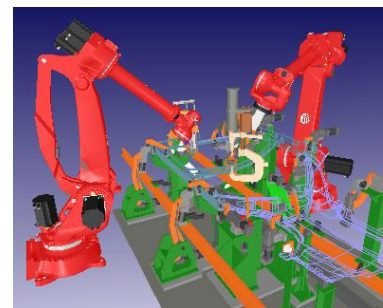
Die ApplicationLib wurde durch neue Beispielzellen erweitert und einige bestehende Beispiele erneuert bzw. aktualisiert. Neben rundum erneuerten Beispielzellen für Punktschweiß- und Lackier-Applikationen, finden Sie neue Beispiele zu Applikationen wie z.B. Auftragen, Bahnschweißen, Handling und Messen.



Painting-Airplane-Wing.cel



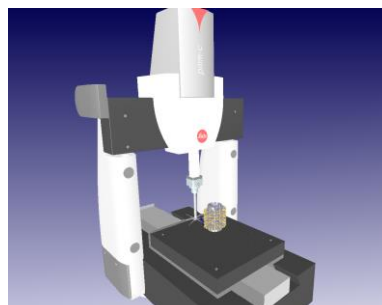
Multi-Dispensing.cel



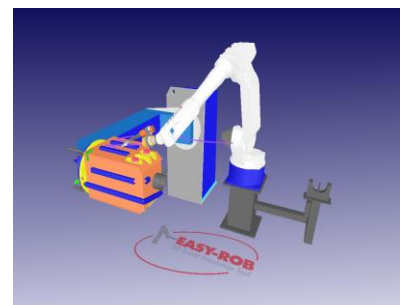
spot_weld_Comau.cel



Flystick-Depalletising.cel



Coordinate-Measuring-VMR.cel



ArcWelding-Chimney-Stove.cel

Kontakt

EASY-ROB 3D Robot Simulation Tool

Stefan Anton
Hauptstr. 42
65719 Hofheim am Taunus
Germany

Tel. +49 (0) 6192 921 70 77
Fax. +49 (0) 6192 921 70 66

Email: contact@easy-rob.com
sales@easy-rob.com

Web: www.easy-rob.com

EASY-ROB Kundenbereich

Online verfügbar: Programm-Updates und Roboterbibliotheken

Web: www.easy-rob.com/special/kundenbereich

Zugangsdaten:

Benutzer:	customer
Passwort:	*****

Eigene Notizen