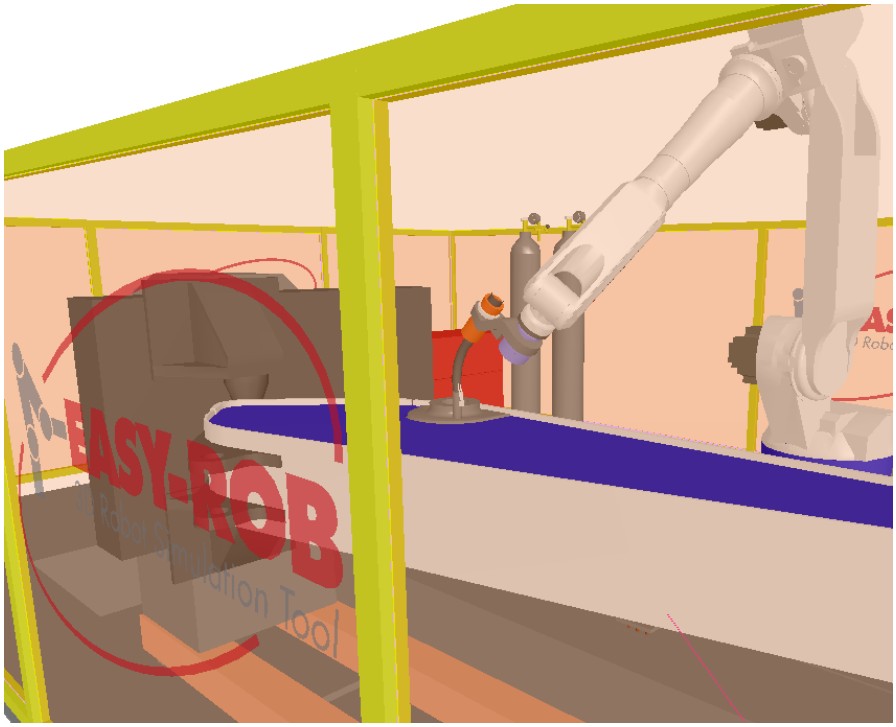
Die neue Version

EASY-ROB™ V6.0



April 2012

Version 1.01

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™

Inhaltsverzeichnis

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ V6.0

Highlights in EASY-ROB™ V6.0

- **EASY-ROB™ vollständig als 64-Bit Version verfügbar**
  Die gesamte EASY-ROB™ Product Suite in der Version 6.0 (**Multi-Program**- und **Single-Robot** Version, **DLL** Version, **Robotics Simulation Kernel** und **Viewer** Version) steht nun ohne Einschränkungen als 64-Bit Version unter Window® 7 64-Bit zur Verfügung.

- **Verbesserter Support mit TeamViewer**
  Damit wir unsere Kunden Online besser unterstützten können, steht Ihnen der EASY-ROB™ TeamViewer Quick Support zur Verfügung. Mit einem Blick auf Ihre oder unsere EASY-ROB™ Sitzung kann so manche Frage schnell beantwortet werden.

- **Neue Robotermodelle**
  Die Roboterbibliothek hat wieder Zuwachs bekommen. Neue Modelle von ABB, Comau, Kuka, Motoman, Stäubli und Universal Robots (UR-5 und UR-10) sind hinzugekommen. Derzeit sind mehr als 450 Roboter vorhanden.

- **Roboter Postprozessoren**
  API Programmierbeispiele sind nun für die Robotersprachen von ABB, Kuka, OTC, Comau, b+m und Fanuc vorbereitet. Geeignete Kunden-Anpassungen erlauben so schnelle Änderungen und Erweiterungen.

- **Voll-Synchro-PTP und SLEW**
  Die Verfahrart Synchro-PTP wurde zum Voll-Synchro-PTP erweitert. Die TCP Bahn ist somit von den programmierten Geschwindigkeiten, Beschleunigungen und Override unabhängig.
  Neue Verfahrart „SLEW“ für asynchrones PTP. Achsen werden in dieser Verfahrart zeitlich nicht synchronisiert.

- **Optimierung**
  Gleiche Geometrien werden geklont und nur noch einmal geladen. Das verkürzt die Ladezeit von Arbeitszellen und Geräten erheblich und spart immense Ressourcen.
  Sämtliche von EASY-ROB™ verwendeten DLLs werden dynamisch gelinkt. Die Namen der benutzerspezifischen DLLs können in der Umgebungsdatei „easy-rob.env“ festgelegt werden.

- **EASY-ROB™ Robotics Simulation Kernel**
  Neue Beispiele für die Verwendung des EASY-ROB™ Robotics Simulation Kernel sollen die Einbindung in eigene Applikationen vereinfachen.
  Zusätzlich zur Lizenzierung durch WibuKey-Dongle und Hardware-Nummer kann der Kernel auch durch den EASY-ROB™ Lizenz Manager lizenziert werden.

- **Neuer Roboter Jog Mode „TCP Base“**
  Der TCP des Roboters kann nun bzgl. seiner Roboterbasis gejogged werden, was mit den Jog-Modi „TCP Tool“ (Werkzeugkoordinaten) „TCP World“ und „Robot Joints“ allen Handverfahrarten am Programmierhandgerät einer Robotersteuerung entspricht.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

- **Pfade und Tag Punkte**
  Viele neue Möglichkeiten Pfade und Tags zu manipulieren. Dazu zählen das Spiegeln von Pfaden, Ausrichten von Tags an eine Achse, Teil-Kopieren von Tagpositionen, Bewegungsumkehr von Pfaden und Redefinition des Work-Objects.
  Weiterhin werden Länge und Winkel des gesamten Pfaden angezeigt, sowie der Abstand zum vorherigen und nachfolgenden Tag.
  Mit „AutoCalc“ werden externe Achswerte, z.B. für einen Positionierer oder einen Dreh-Kipptisch, in Abhängigkeit der Pfadlänge automatisch berechnet. Diese Werte können anschließend optimiert werden.

- **Neuer Kinematik Typ: JET Roboter (Portalroboter)**
  JET Robotern mit der ID 127 (bzw. 128 für A2A3 Kopplung) mit der seriellen Struktur TyRyy:Rxyx oder TyRyy:Rzyz mit 4 Konfigurationen werden unterstützt und können so einfach auch vom Kunden erstellt werden. (z.B. KR 30 JET, KR 60 JET)
  Zwei- und Drei-achsige Gantry-Kinematiken (ID 133) lassen sich in den Kombinationen Txyz, Tyxz, Tzxy, Tzyx, Txzy, Tyzx bzw. Txz, Tyz, Tzx, Tzy, Txy, Tyx erstellen.

- **Neue Roboter-Attribute**
  Für aktive- und passive Achsen von Robotern, Kinematiken bzw. Geräten (Devices) können Namen vergeben werden, z.B. Hubachse, Querantrieb oder Achse_1 anstatt Joint_1.
  Auch lassen sich Namen für die Konfigurationen vergeben, z.B. für Kuka S'B010 oder Fanuc NUT anstatt Config_1.
  Neben dem Turn-Intervall können auch Turn-Offsets für jede Achse definiert werden, was für den Abgleich mit der realen Roboter-Steuerung unablässig ist.

- **Export von Pfaden in native Roboterprogramme**
  Der generische Export von Pfaden in native Roboterprogramme ist für die Steuerungstypen von ABB, KUKA, COMAU, Fanuc, b+m und OTC erweitert worden. Weitere werden je nach Kundenwunsch eingebunden.

- **Kollision**
  Der neue Kollisions-Algorithmus „PQP“ erlaubt es Toleranzen zu definieren. Demnach wird Kollision angezeigt wenn zwei Körper sich auf einen minimalen Abstand nähern. Dieser Toleranzwert kann nun für jeden einzelnen Körper individuell vorgegeben werden.

- **Neue ERPL und ERCL Befehle**.
  SLEW, SLEW_REL, SLEW_AX, SLEW_AX_REL für asynchrones PTP
  “ERC GRAB_TO” zum Re-Attachen von Geräten an andere Geräte
  “ERC COLLISION DISTANCE …“ zur Definition des Kollisions-Toleranzwertes einzelner Geometrien

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

- **API Application Program Interface, die Klasse ER_CAPI**
  Für individuelle Produktanpassungen und spezielle Lösungen sind viele neue API Funktionen hinzugekommen. Diese ermöglichen es beispielsweise EASY-ROB™ aus einer eigenen Applikation anzusteuern bzw. bidirektional Daten auszutauschen. Die Methoden-Klasse ER_CAPI dient als Schnittstelle für die EASY-ROB™ **Multi-Program** und EASY-ROB™ **DLL** Version sowie für die Erweiterungen **API-INV**, **API-IPO**, **API-DYN**, **API-UserDLL**, **API-PostProc** und **API-Sensors**.

- **Sonstiges**
  - Teach-Window mit neuem Dialog mit sämtlichen Bewegungsbefehlen
  - Device Manager Dialog nun skalierbar
  - AVI-Recorder mit weiteren Auflösungen
  - 3D Space Mouse mit einstellbarer Empfindlichkeit und Schwellwert in Umgebungsdatei
  - Neue Parser-Funktionen mit Zugriff auf kinematische Roboterlängen und mehr
  - Erweiterter Status Output für die Ausgabe von z.B. Achswerten in jedem Simulationsschritt

Die neue Version steht für alle Kunden mit einer gültigen Lizenz oder einem Softwarepflegevertrag für EASY-ROB™ V6.0 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

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ Version 6.0

Nach 12 Jahren EASY-ROB gibt es nun die Version 6.0 mit vielen Neuerungen und Verbesserungen.

Sicherlich besteht nach wie vor der Wunsch vieler Kunden und vor allem Interessenten nach Vereinfachung. So möchte doch jeder bei der simulationsgestützten Planung einer Roboterarbeitszelle möglichst schnell eine zuverlässige Aussage über Machbarkeit und Taktzeit erhalten. Gleichzeitig steigen jedoch Komplexität und Anforderungen der Anlage. Insofern stehen wir wie immer vor dem Dilemma: „Einfacher mit weniger Funktionalität oder zum Teil komplexer aber sehr viel ist möglich“. Wir haben uns bewusst für den etwas steinigeren Weg entschieden und wollen unseren Kunden und OEM Partnern mit der EASY-ROB™ Product Suite weiterhin viel Funktionalität zur Verfügung stellen. Dazu zählt die konsequente Weiterentwicklung der Programmierschnittstellen (ER_CAPI), der EASY-ROB™ DLL Version sowie des EASY-ROB™ Robotics Simulation Kernels.

Nach unserer Auffassung ist ein gewisses Maß an Robotics Know-How Grundvoraussetzung für die effektive Bedienung von EASY-ROB™. Nur so kann verstanden werden, welche Roboter-Einstellungen Auswirkungen auf Bahn, Orientierungsverhalten und Taktzeit haben. Hinzukommen sicherlich dreidimensionales Vorstellungsvermögen sowie mathematische Fähigkeiten um die Bedeutung von 3D Koordinaten (z.B. Euler Winkel) und Transformationen im Raum besser zu verstehen. Benutzer der API bringen natürlich noch Programmierkenntnisse in C/C++ mit.

Um das Problem zu lösen, bieten wir zusätzlich zur EASY-ROB™ Schulung nun auch eine eintägige Robotics Schulung an.

Trotzdem ist es natürlich immer unser Ziel die Bedienung von EASY-ROB™ zu vereinfachen.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Support mit TeamViewer

Damit wir unsere Kunden Online besser unterstützten können, steht Ihnen der EASY-ROB™ TeamViewer Quick Support zur Verfügung. Mit einem Blick auf Ihre oder unsere EASY-ROB™ Sitzung kann so manche Frage schnell beantwortet werden.

Unterstützung mit TeamViewer Quick Support

Laden Sie sich TeamViewer QJ herunter.

- http://www.easy-rob.com/service/beratung.html

Das Programm „teamviewerqs_er_de.exe“ kann ohne Installation und ohne Administratorrechte gestartet werden und erlaubt es uns Sie spontan zu unterstützen.

Präsentation mit TeamViewer Quick Join

Laden Sie sich TeamViewer QJ herunter

- http://www.easy-rob.com/service/praesentation.html

Das Programm „teamviewerqj_er_de.exe „ kann ohne Installation und ohne Administratorrechte gestartet werden und erlaubt es uns Ihnen unsere Produkte zu präsentieren.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

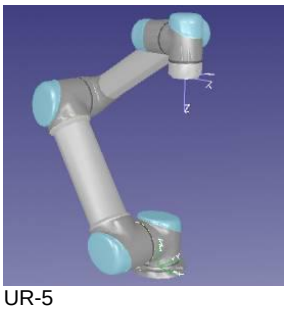
Neue Robotermodelle

Die Roboterbibliothek hat wieder Zuwachs bekommen. Neue Modelle von ABB, Comau, Kuka, Motoman, Stäubli und Universal Robots (UR-5 und UR-10) sind hinzugekommen.

Derzeit sind mehr als **450 Roboter** vorhanden

Universal Robot

UR-5
UR-10


UR-5


UR-10

ABB

IRB-6640ID_170_275
IRB-6640ID_200_255
IRB-6640_130_320
IRB-6640_180_255
IRB-6640_185_280
IRB-6640_205_275
IRB-6640_235_255


IRB-6640ID_170_275


IRB-6640_180_255

KUKA

KR-90-R2700-pro
KR-90-R3100-extra
KR-120-R2500-PRO
KR-120-R2900-EXTRA
KR-150-R2700-EXTRA
KR-180-R2500-EXTRA
KR-210-R2700-EXTRA
KR-90-R3700-K-Prime
und weitere


KR-300-R2500-ULTRA


KR-90-R3100-extra

COMAU

NJ-130-2,6
NJ-110-3,0


NJ-130-2,6


NJ-110-3,0

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Staubli Roboter

TX200-HB-L
TX200-L-60
TX200-100
TP80

TX200L-HB-L

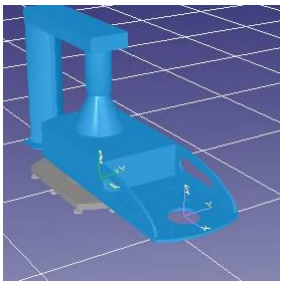
TP80

Motoman

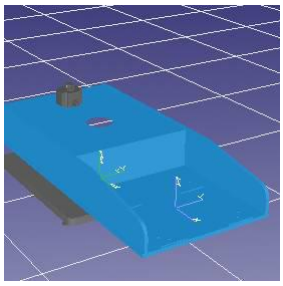
MA1900
MA1800
MA1400
VST-600
VST-1500

MA1900

MA1400



VST-1500



VST-600

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Roboter Postprozessoren (API)

API Programmierbeispiele sind nun für die Robotersprachen von ABB, Kuka, OTC, Comau, b+m und Fanuc vorbereitet. Geeignete Kunden-Anpassungen erlauben so schnelle Änderungen und Erweiterungen.

Die Option API-Post-Proc erlaubt es eigene Post-Prozessoren für EASY-ROB™ zu entwickeln. In einem Visual Studio 2008 Beispielprojekt „er_post.sln“ sind Postprozessoren für ABB, Kuka, b+m, Fanuc, OTC und Fanuc vorbereitet. Die erzeugte DLL „er_post.dll“ wird dynamisch gelinkt.

In ERPL wird der Postprozessor mit dem Befehl

„ERC POST_PROCESS LANGUAGE_KEY flnname“

Für LANGUAGE_KEY kann z.B. KUKA, OTC oder Fanuc gewählt werden. Diese Keys können beliebig erweitert werden.

Im Programmierbeispiel wird die Klasse „`ER_CPost_Base`“ abgeleitet, so dass zu Beginn nur die wesentlichen (abstrakten) Methoden überschrieben werden müssen.

```
class ER_CPost_Base
{
public:
   ER_CPost_Base(void);
   virtual ~ER_CPost_Base(void);
public:
   // abstract methods
   virtual int pp_Header(char *fln)=0;
   virtual int pp_Feed(void)=0;
   virtual int save_Tag_PTP_Motion(char *path_name,char *tag_name,char *tool_name,char
*wobj_name)=0;
   virtual int save_Tag_LIN_Motion(char *path_name,char *tag_name,char *tool_name,char
*wobj_name)=0;
   virtual int save_Tag_CIRC_Motion(char *path_name,char *via_tag_name,char *tag_name,char
*tool_name,char *wobj_name)=0;

   virtual int save_HOME_Motion(float *q,double *q_ext,char *tool_name,char *wobj_name)=0;
   virtual int save_PTP_AX_Motion(float *q,double *q_ext,char *tool_name,char *wobj_name)=0;
   virtual int save_PTP_Motion(float *x,double *q_ext,char *tool_name,char *wobj_name)=0;
   virtual int save_LIN_Motion(float *x,double *q_ext,char *tool_name,char *wobj_name)=0;
   virtual int save_CIRC_Motion(float *x,float *x_via,double *q_ext,double *q_ext_via,char
*tool_name,char *wobj_name)=0;

   virtual int frame_to_vec_CTRL(float *v, frame *T)=0;

   // virtual methods, overwriting optional
   virtual int pp_Export(char *fln_prg_line,int pp_mode);
   virtual int pp_Cyclic (char *prg_line);
   virtual int save_BASE(frame *Tbase);
   virtual int save_WAIT(float sec);
   virtual int save_TOOL(frame *Ttool);
   virtual int save_SPEED_PTP(float vq);
   . . .
   virtual int save_NATIVE(char *prg_line);
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Roboter Postprozessoren (API)

```
public:
   char *split_filename(char *fln);          // splits status_fln into drive path file ext
   int append_file(FILE *fp,char *path,char *afln,char *comment=NULL);
                              // Append content from file 'path+afln' to current file stream fp
   int frame_to_vec_ABB(float *v, frame *T);        // predefined function for frame_to_vec_CTRL()
   int frame_to_vec_KUKA(float *v, frame *T);       // predefined function for frame_to_vec_CTRL()
   char *generate_zone_string(float zone_value);

protected:
   FILE * fp;                                // 1st File stream
   FILE * fpd;                               // 2nd File stream
   FILE * pfp;                               // points to fp or fpd

   char status_fln[HS_MAXSTR];               // complete file name
   char status_path[HS_MAXSTR];              // drive and path
   char status_fname[HS_MAXSTR];             // file name without extension
   char status_ext[_MAX_EXT];                // file extension

   char program_line[HS_MAXSTR];             // current program line
   char program_key[HS_MAXSTR];              // current program key, first word in program line

   int num_dofs;                             // number of robot joints
   . . .
}; // class ER_CPost_Base
```

Mit der PostProzessor API können auch individuelle Sprachen herausgeschrieben werden, je nach Anwendung und Steuerung.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

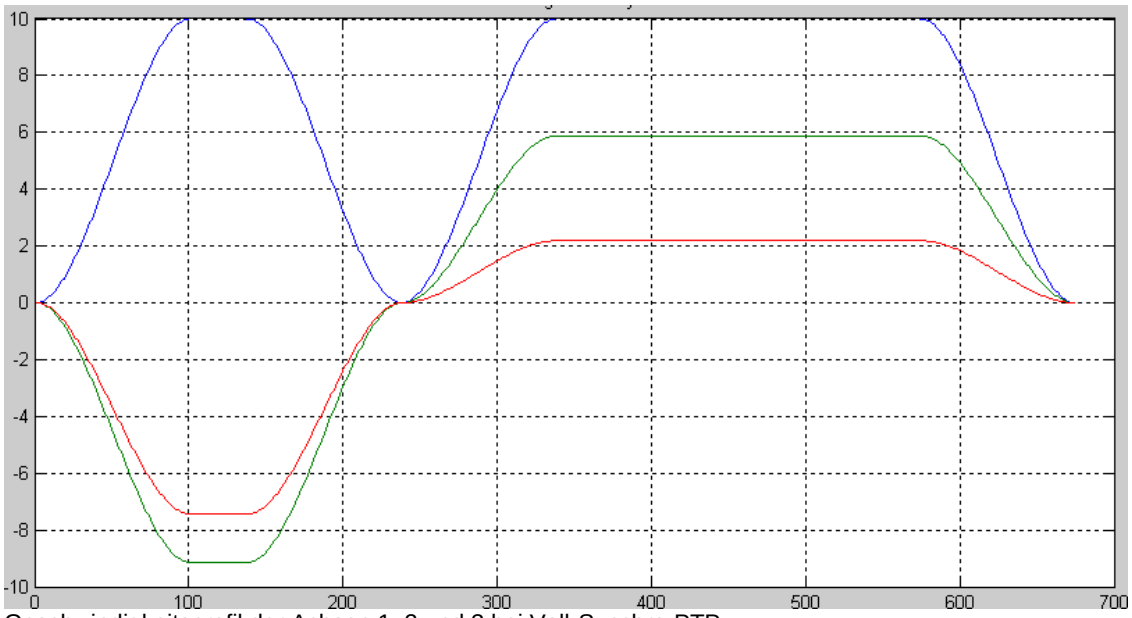
Voll-Synchro-PTP und SLEW Motion

Die Verfahrart Synchro-PTP wurde zum Voll-Synchro-PTP erweitert. Die TCP Bahn ist somit von den programmierten Geschwindigkeiten, Beschleunigungen und Override unabhängig.
Neue Verfahrart „SLEW“ für asynchrones PTP. Achsen werden in dieser Verfahrart zeitlich nicht synchronisiert

Voll-Synchro-PTP

Beim Synchronen PTP fangen alle Achsen zur gleichen Zeit mit ihrer Bewegung an und beenden diese auch zur gleichen Zeit. Die so genannte Leitachse oder dominierende Achse gibt dabei die Zeit der gesamten Bewegung an. Die Leitachse ist in der Regel die Achse mit der größten Zeit. Die anderen Achsen werden zeitlich angepasst.

Beim Voll-Synchronen-PTP (auch phasensynchrones PTP genannt) wird zusätzlich gewährleistet, dass die Beschleunigung- und Bremsphasen aller beteiligten Achsen der Leitachse angepasst werden.



Geschwindigkeitsprofil der Achsen 1, 2 und 3 bei Voll-Synchro-PTP
Achse1 (blau) ist Leitachse, nur sie erreicht die programmierten Achsgeschwindigkeit mit v = 10°/s bei a = 10°/s²
Die Achsen 2 (grün) und 3 (rot) werden verlangsamt.
Zelle: „PTP-SLEW.cel“ mit status output file „ER431_2_PTP.dat“

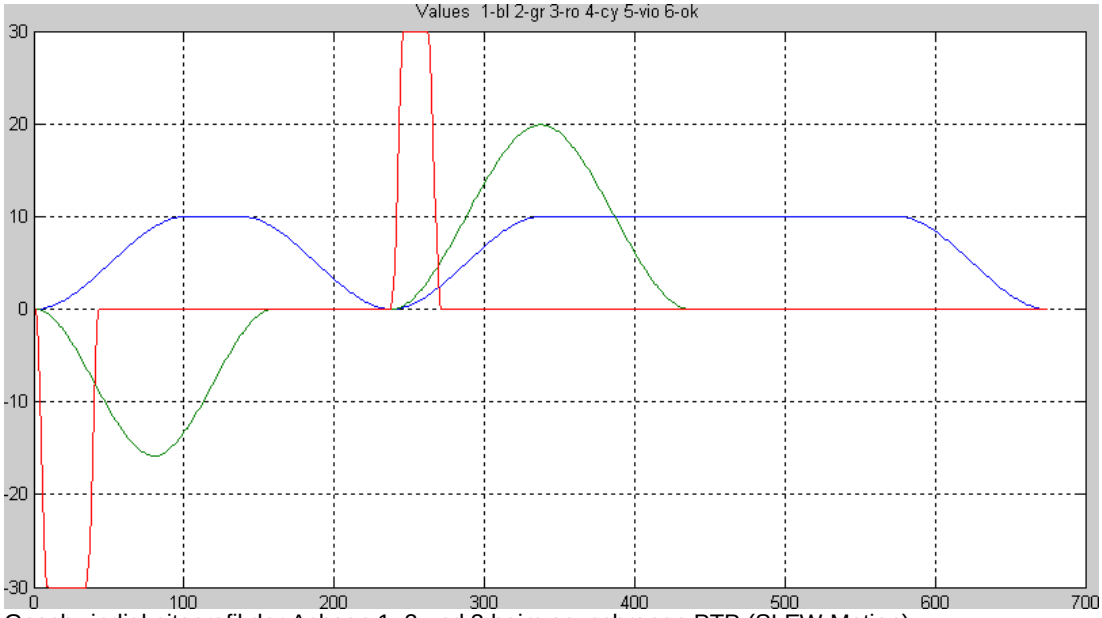
Der entscheidende Vorteil bei Voll-Synchro-PTP besteht darin, dass die sich ergebende TCP Bahn des Roboters unabhängig von den programmierten Achsgeschwindigkeiten und Achsbeschleunigungen ist.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Asynchrones PTP (SLEW)

Beim aynchronen PTP (SLEW Motion) fangen alle Achsen zur gleichen Zeit mit ihrer Bewegung an. Je nach programmierte Geschwindigkeit, Beschleunigung und Wegstrecke erreichen die Achsen zu unterschiedlichen Zeiten ihre Zielposition. Es gibt also keine Leitachse, bzw. keine Synchronisation mit den anderen Achsen.

Diese „neue“ Verfahrart SLEW eignet sich eher für „sonstige“ Geräte anstatt für Roboter.



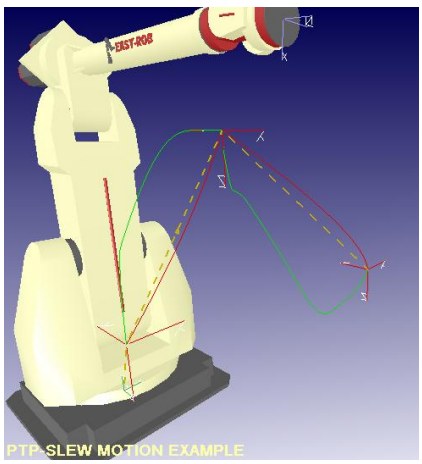
Geschwindigkeitsprofil der Achsen 1, 2 und 3 beim asynchronen PTP (SLEW Motion)
Alle Achsen erreichen ihre programmierten Achsgeschwindigkeiten mit 10°/s, 20°/s und 30°/s.
Die Achsen 2 (grün) und 3 (rot) erreichen vorzeitig ihre Endposition. Die Achse 1 (blau) ist am langsamsten.
Zelle: „PTP-SLEW.cel“ mit status output file „ER431_2_SLEW.dat“

Beispielzelle: „PTP-SLEW.cel“

Das Programm „PTP-SLEW-ER431-2.prg“ erzeugt die zwei Status Output Dateien „ER431_2_PTP.dat“ und „ER431_2_SLEW.dat“, die mit den MATLAB® files „ptp_slew.m“ und „show_ax.m“ grafisch dargestellt werden. Aufruf ptp_slew(3), 3 = Anzahl der dargestellten Achsen, hier 1,2 und 3.

TCP-Spur:

PTP → rot
SLEW → grün



Vergleich PTP- und SLEW Motion “

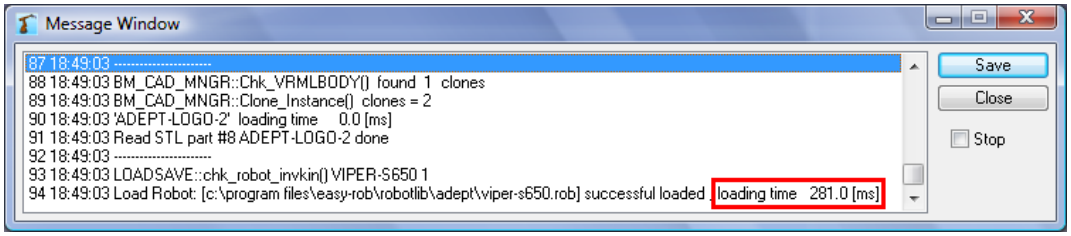
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Optimierungen

Verkürzung der Ladezeit durch Klonen gleicher Geometrien

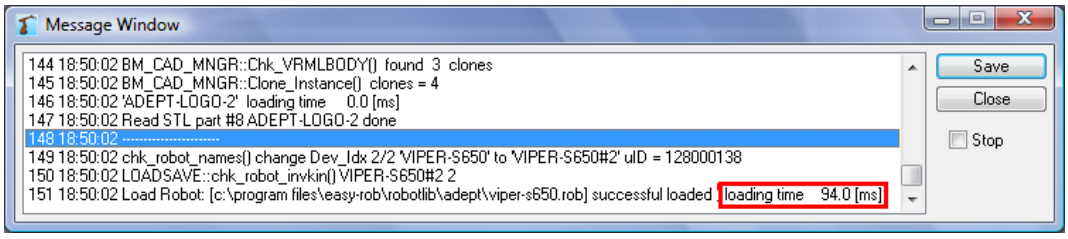
Gleiche Geometrien werden nicht mehr doppelt geladen sondern „geklont“. Ein wesentlicher Vorteil besteht in der Steigerung der Performance, wodurch ein schnelleres Laden von Robotern, die gleiche Geometrien beinhalten und eine niedrige Speicherausnutzung (gerade bei komplexen Geometrien) erreicht wird.

1. Öffnen des Message-Windows (Ctrl + M) und laden eines Roboters



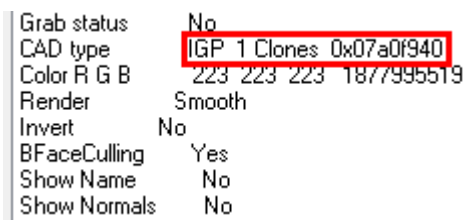
Das erstmalige Laden des Roboters VIPER-S650 benötigt eine Ladezeit von etwa 281 ms.
(Beachten Sie, dass es sich bei der Angabe der Ladezeit um einen Wert handelt, der von System zu System variiert. Dieses Beispiel dient zur Veranschaulichung der Ladezeit-Ersparnis)

2. Laden eines zweiten (identischen) Roboters



Wird nun ein zweiter Roboter desselben Typs geladen, so verkürzt sich die Ladezeit auf etwa 94 ms. Dabei entfällt durch das Klonen die gesamte Ladezeit von bereits vorhandenen Geometrien.

Hinweis: Werden identische Geometrien gefunden, so werden Name und Anzahl der erstellten Klone im 3D-CAD Window angezeigt.



Die geladenen Geometrien des Roboters können im 3D-CAD Window angewählt und die dabei vorhanden Anzahl der Klone in der Zeile „CAD-type“ abgelesen werden.

Hinweis: Attribute wie z.B. Name, Render-Typ und Farbe, Offset-Position, Collision-Tolerance usw. werden gedoppelt, da jede Geometrie nachwievor ihre eigenen Attribute benötigt.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Dynamisches Laden von DLLs

Sämtliche von EASY-ROB™ verwendeten DLLs werden dynamisch gelinkt. Die Namen der benutzerspezifischen DLLs können in der Umgebungsdatei „easy-rob.env“ festgelegt werden.

EASY-ROB™ läst sich so mit der individuellen Lösung starten.

Umgebungsdatei: „easy-rob.env“

```
! User defined name for inverse kinematics, er_kin.dll
ER_KIN_DLL ER_KIN.DLL
!
! User defined name for motion planning and execution, er_ipo.dll
ER_IPO_DLL ER_IPO.DLL
!
! User defined name for dynamics and control, er_dyn.dll
ER_DYN_DLL ER_DYN.DLL
!
! User defined name post processor, er_post.dll
ER_POST_DLL ER_POST.DLL
!
! User defined name sensor, er_sensor.dll
ER_SENSOR_DLL ER_SENSOR.DLL
!
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

EASY-ROB™ Robotics Simulation Kernel

Der EASY-ROB™ Simulation Kernel ist eine Entwickler Version zur Integration in eigene Applikationen. Der Kernel übernimmt sämtliche Berechnungen wie Vorwärts- und Inverse Koordinatentransformation, Bewegungsplanung und -ausführung (PTP, LIN und CIRC) für alle verfügbaren Robotertypen. Zur Ansteuerung werden ausschließlich C/C++ API-Funktionen/Services für die Roboter-Funktionalität zur Verfügung gestellt. Die eigene Applikation übernimmt die 3D Visualisierung, sowie die Verwaltung sämtlicher Geometrien und Handles der geladenen Kinematiken. Der EASY-ROB™ Simulation Kernel liefert für jede geladene Kinematik ein Handle zurück.

Neue Beispiele für die Verwendung des EASY-ROB™ Robotics Simulation Kernel sollen die Einbindung in eigene Applikationen vereinfachen.

Zusätzlich zur Lizenzierung durch WibuKey-Dongle und Hardware-Nummer kann der Kernel auch durch den EASY-ROB™ Lizenz Manager lizenziert werden.

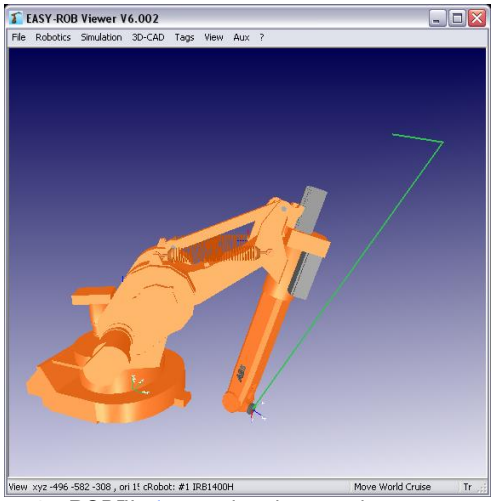
Zu den EASY-ROB™ Kernel files gehören die Dateien

- version.txt // Version
- EasySimKernel.dll // Windows Dll
- EasySimKernel.lib // Library für Linker
- EasySimKernel.def // def Datei falls benötigt
- er_Kernel_main.h // Header files , Deklaration von Typen , Prototypen, etc.
- ipo_extax.h // Header files,
- er_wibukey.dll // für WibuKey USB Dongle Lizenzierung
- EasySimKernelx64.dll // Windows Dll, 64-Bit Version
- EasySimKernelx64.lib // Library für Linker, 64-Bit Version
- er_wibukeyx64.dll // für WibuKey USB Dongle Lizenzierung, 64-Bit Version

Kinematik Beispiel

In diesem einfachen Kinematik-Beispiel wird die TCP Position in jedem Schritt so lange verändert, bis der Aufruf der inversen kinematischen Lösung einen Fehler zurückgibt.

In diesem Fall sind die Verfahrbereiche der Achse 2 verletzt. Ein weiterer Fehler wäre z.B. die nicht Erreichbarkeit der vorgegebenen Position.



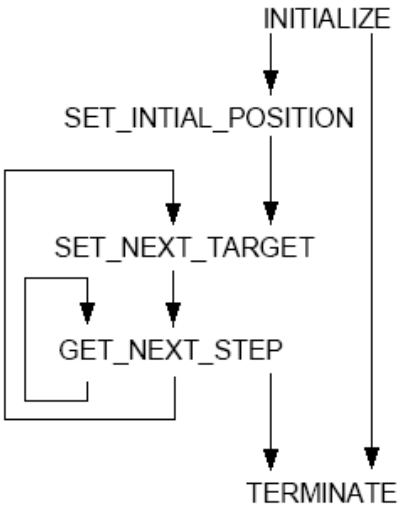
EASY-ROB™ Viewer mit Roboter und Programm

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

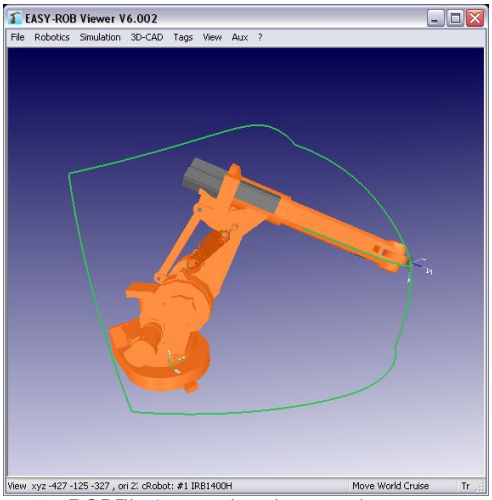
Motion Planner Beispiel für PTP, LIN, CIRC

Im Motion Planner Beispiel werden einige Zielpositionen in PTP, LIN oder CIRC angefahren.

Im Beispiel wird gezeigt wie Geschwindigkeiten etc. vorgegeben und Fehler abgefangen werden.



Prinzipielle RRS-Services (Quelle: Fhg-IPK-Berlin)



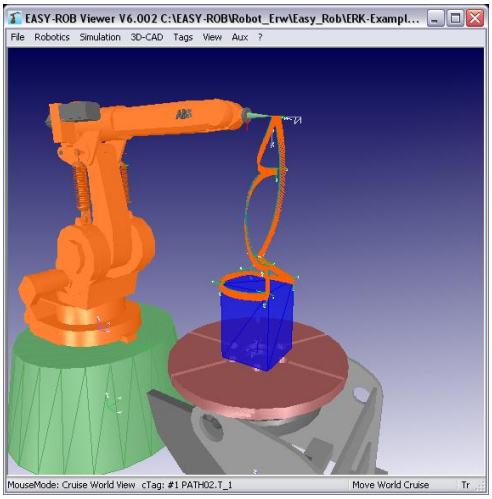
EASY-ROB™ Viewer mit Roboter und Programm

Motion Planner Beispiel mit synchronisiertem Positionierer

In diesem Beispiel wird die Roboterbewegung mit dem externen Positionierer synchronisiert.

Die Zielpositionen sind bzgl. des Tisches bzw. Bauteil definiert.

Im Beispiel wird gezeigt wie Geräte im Raum platziert, miteinander verknüpft, synchronisiert und zu den Targets auch externe Achswerte vorgegeben werden.



EASY-ROB™ Viewer mit Roboter und Programm

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Module und Optionen

Der EASY-ROB™ Robotics Simulation Kernel ist in Module aufgeteilt, so sukzessive Optionen hinzugebucht werden können.

Das Basismodul besteht aus „erk01 - ERK Kinematics - Single Robot“ und beinhaltet Vorwärts- und Inverse Koordinatentransformation.

Soll der Kernel mehrere Kinematiken gleichzeitig verwalten ist Basismodul „erk03 - ERK Multi-KIN“ erforderlich. Optional sind wie bei EASY-ROB™. die Roboterbibliotheken „opk*“ von ABB, Kuka, Fanuc, Motoman, Staubli und PKM.

Für den Motion Planner ist das Basismodul „ERK Motion Planner“ erforderlich. Damit kann eine Kinematik in den Verfahrarten PTP, SLEW, LIN und CIRC bzgl. Roboterbasis verfahren werden.

Der Motion Planner kann mit den Motion Planner Optionen „opm*“ je nach Bedarf erweitert werden. So wird z.B. für die werkzeugführende Bewegung die Option „opm01“ benötigt.

Für die Einbindung in ihre Applikation bieten wir gerne unsere Unterstützung an.

Item No.	Product

ERK Basic Moduls

erk01	ERK Kinematics - Single-Robot
erk02	ERK Motion Planner - PTP, LIN, CIRC - Werkstückführend - Auxiliary Axis
erk03	ERK Multi-KIN

Item No.	Product

Options Kinematics

opk01	ERK Kuka
opk02	ERK Staubli
opk03	
opk04	ERK Tricept
opk05	ERK ABB
opk06	ERK Motoman
opk06	ERK Fanuc
opk07	
opk08	
opk09	ERK PKM-Delta

Options Motion Planner

opm01	ERK Werkzeugführend
opm02	ERK Positioner
opm03	ERK Conveyor
opm04	ERK Trackmotion
opm05	ERK Robot
opm06	ERK Tracking Window
opm07	ERK KIR

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Neuer Roboter Jog Mode „TCP Base“

Der TCP des Roboters kann nun bzgl. seiner Roboterbasis gejogged werden, was mit den Jog-Modi „TCP Tool“ (Werkzeugkoordinaten) „TCP World“ und „Robot Joints“ allen Handverfahrarten am Programmierhandgerät einer Robotersteuerung entspricht.

Joggen des TCPs in Werkzeugkoordinaten

NEU: Joggen des TCPs bzgl. seiner Roboterbasis

Joggen des TCPs in Weltkoordinaten

Joggen der Roboterachsen

NEU: Zoomen auf

- Welt ()
- cRobot (TCP Base)
- cBase (Robot Base)
- cTcp (TCP Tool)
- cBody ()
- cTag (Sel Tag)

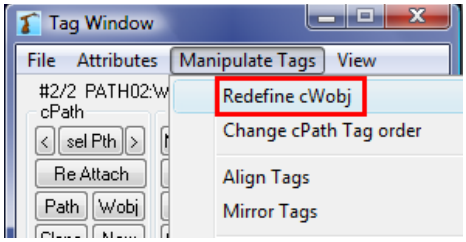
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Pfade und Tag Punkte manipulieren

Viele neue Möglichkeiten Pfade und Tags zu manipulieren. Dazu zählen das Spiegeln von Pfaden, Ausrichten von Tags an eine Achse, Teil-Kopieren von Tagpositionen, Bewegungsumkehr von Pfaden und Redefinition des Work-Objects.
Weiterhin werden Länge und Winkel des gesamten Pfaden angezeigt, sowie der Abstand zum vorherigen und nachfolgenden Tag.
Mit „AutoCalc“ werden externe Achswerte, z.B. für einen Positionierer oder einen Dreh-Kipptisch, in Abhängigkeit der Pfadlänge automatisch berechnet. Diese Werte können anschließend optimiert werden

Redefine cWobj Position

Mit Hilfe von „Redefine cWobj“ wird die Workobject-Position verändert, ohne dass die Tagpunkte ihre Position im Raum ändern. Dies ist vor allem dann vorteilhaft, wenn das Setzen von Tag-Punkten oder Planen von ganzen Pfaden durch ungünstige Lage der Workobject-Position erschwert wird.

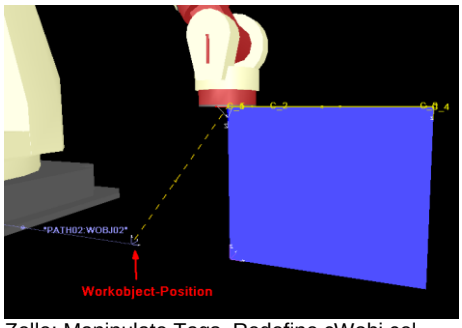


Öffnen Sie das Tag Window und wählen Sie unter *Manipulate Tags > Redefine cWobj*

Der Frame Dialog öffnet sich. Dort können Sie die neue Lage der Workobject-Position eingeben.

Eine einfaches Beispiel soll den Nutzen der Funktion „Redefine cWobj“ verdeutlichen

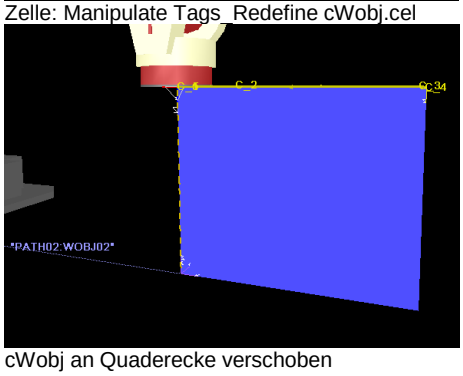
Die Workobject-Position liegt außerhalb des blauen Bauteiles. Bezugssystem für die Tag-Punkte an der Geometrie ist das cWobj. Werden weitere Tag-Punkte an der Würfel-Geometrie angebracht werden, so kann die ungünstige Lage des cWobj, die Planung erschweren, da bei der Angabe der Tag-Punkt-Koordinaten das cWobj das Bezugssystem ist, und nicht z.B. der Koordinatenursprung der Geometrie



Zelle: Manipulate Tags_Redefine cWobj.cel

Wird der Ursprung des cWobj auf eine der 8 Quaderecken gesetzt, so erleichtert dies die Planung von Tag-Punkten an der Geometrie

Eine weitere typische Anwendung dieser Funktion ist nach dem Reattachen von Pfaden.

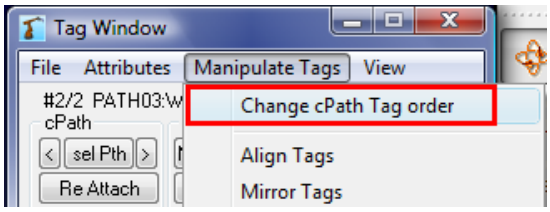


cWobj an Quaderecke verschoben

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

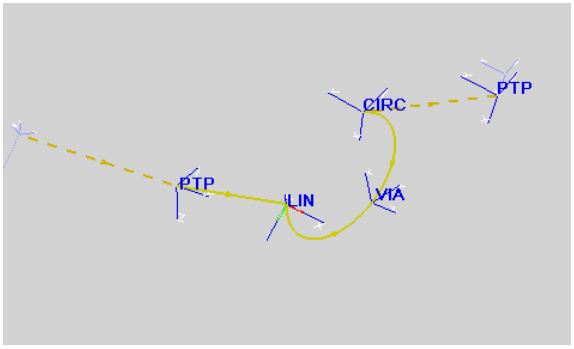
Change cPath Tag order

Ändert man die Anfahrrichtung einer vorgegebenen Tag-Reihenfolge (z.B. Pfad „rückwärts abfahren“), so muss dabei der Bewegungstyp (PTP, LIN; VIA; CIRC), mit dem der jeweilige Tag-Punkt angefahren wird, in der Regel geändert werden.



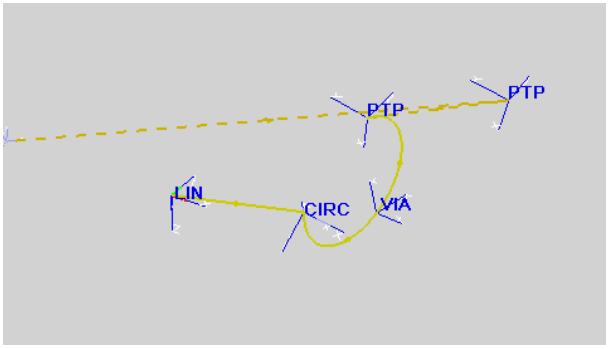
Über Manipulate Tags > Change cPath Tag order kann die Anfahr-Reihenfolge der Tag-Punkte und die damit verbundenen Bewegungstypen mit einem Mausklick geändert werden.

Die folgende Bewegungsabfolge soll die Problematik verdeutlichen.



Manipulate Tags_Change Tag Order.cel

Originaler Pfad: **PTP → LIN → VIA → CIRC → PTP**



Den originalen Pfad vorab klonen !!!

Neuer Pfad: **PTP-NEU → PTP → VIA → CIRC → LIN**

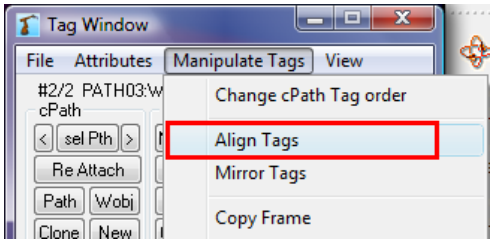
Da beim „Change cPath Tag order“ der erste Tag hinausgeschoben und der letzte Tag ergänzt wird, werden diese immer durch die Verfahrart PTP ergänzt.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Align Tags – Ausrichten von Tag-Punkten

Beim Ausrichten von Tag-Punkten wird die x-, y- oder z-Achse der selektierten Tag-Punkte an eine Achse des ausgewählten „Referenz-Tag-Punktes“ ausgerichtet.

Dies dient dazu, die Achsbewegungen des Roboters zu reduzieren, um dadurch eine höhere Präzision des Bewegungsablaufs und damit eine verbesserte Qualität zu erreichen.

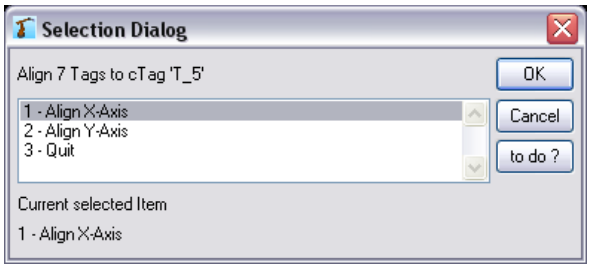


Über *Manipulate Tags* > *Align Tags* können die ausgewählten Tag-Punkte ausgerichtet werden.

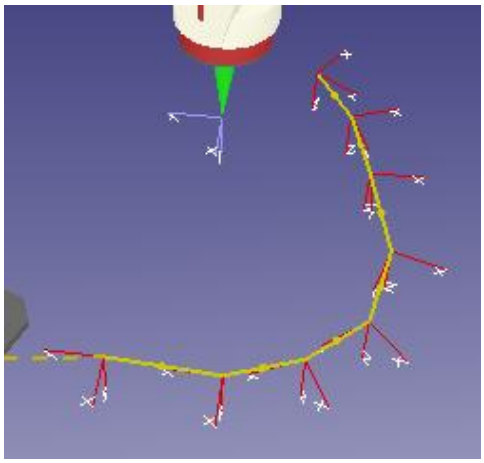
Beachten Sie:
Beim Selektieren der Tag Punkte entscheiden Sie *welche Tag-Punkte* ausgerichtet werden sollen und *welcher Tag-Punkt* als Referenz Tag dient.

Die Reihenfolge ist dabei entscheidend: Der zuletzt in der Liste angewählte Tag-Punkt dient als „Referenz-Tag-Punkt“, an dem alle anderen Tag-Punkte ausgerichtet „aligned“ werden.

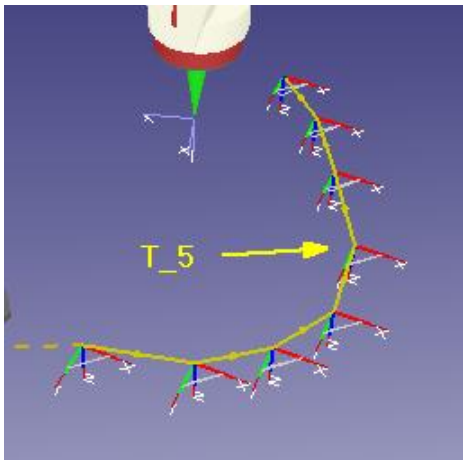
Wählen Sie zuletzt noch aus, welche Achse der Tag-Punkte ausgerichtet werden soll.



Nur die x- und y-Achse stehen zur Auswahl, da die z-Achse als Approach Achse gewählt ist.
Menu → Attributes → Tag Approach Direction



Manipulate Tags_Align_Tags.cel

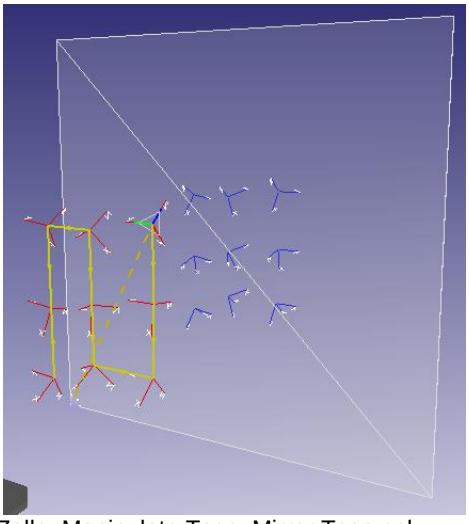


Die x-Achsen der Tag-Punkte wurden an die x-Achse des Referenz-Tag-Punktes „T_5“ ausgerichtet.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Mirror Tags - Spiegeln von Tag-Punkten

Ein oder mehrere Tag Punkte eines Pfades lassen sich an Ebenen spiegeln.

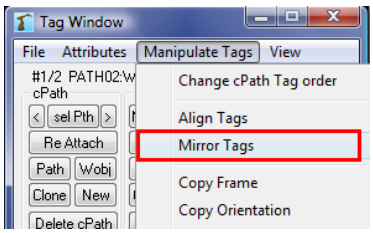


Zelle: Manipulate Tags_Mirror Tags.cel

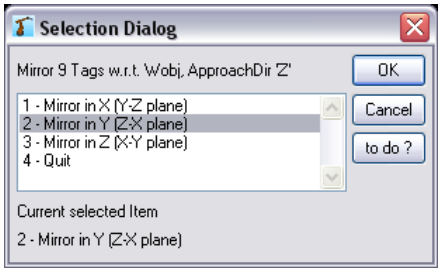
Das Bezugssystem wird durch das cWobj-Koordinatensystem festgelegt.
Oben zu sehen: Die zu spiegelnden Tag-Punkte in Rot und die gespiegelten Tag-Punkte in der Farbe Blau. Die weiße Fläche soll die Spiegelebene verdeutlichen.

Öffnen Sie das Tag-Window und wählen Sie die Tag Punkte des zu spiegelnden Pfades aus.

Tipp: Klonen Sie den Pfad vorab



Über *Manipulate Tags* > *MirrorTags* können die ausgewählten Tag-Punkte gespiegelt werden.



Bevor die Tag-Punkte gespiegelt werden können, müssen Sie auswählen, welche Fläche als „Spiegelebene“ dienen soll.

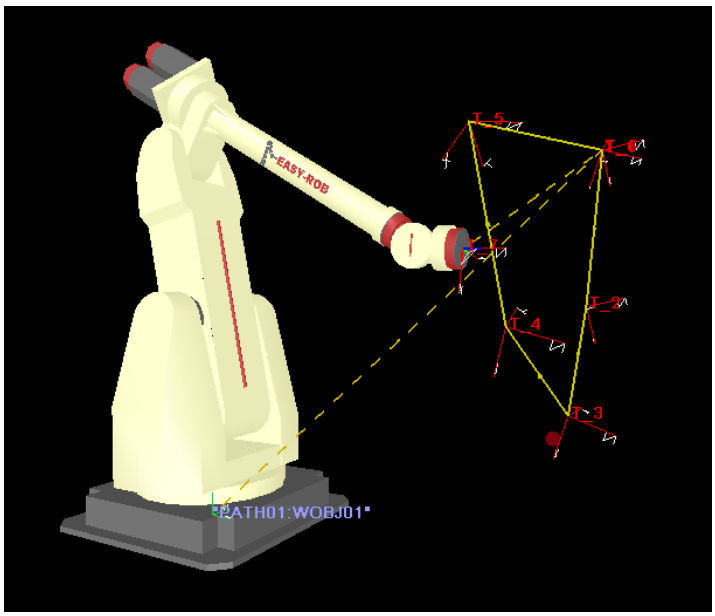
Mirror in X *(*Spiegelebene *wird von Y-Z-Achse aufgespannt)*
Mirror in Y *(*Spiegelebene *wird von Z-X-Achse aufgespannt)*
Mirror in Z *(*Spiegelebene *wird von X-Y-Achse aufgespannt)*

Beachten Sie: Die Approach-Direction spielt eine wesentliche Rolle beim Spiegeln. Die Approach-Achse (in diesem Fall die Z-Achse) ist dominant und wird beim Spiegeln sozusagen „übernommen“. Sie entscheidet darüber wie das gespiegelte Koordinatensystem ausgerichtet wird. (Erhaltung des Koordinaten-Rechtssystems). Bei Approach-Achse z, werden z- und x-Achse gespiegelt und die y-Achse ändert ihre Richtung. Ist das nicht gewünscht, müssen alle gespiegelten Tag anschließend um 180° um die z-Achse rotiert werden.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Kopieren von Tag-Punkt-Position/Orientierung

Jede Tag-Position (Tag-Frame) besteht aus der x-, y- und z-Position und aus einem Orientierungsteil. Teile dieses Referenz-Tags können nun auf selektierte Tags übertragen werden.



Manipulate Tags_Copy Tags.cel

Dabei können Sie zwischen 6 verschiedenen Copy-Befehlen wählen:

1. Copy Frame
2. Copy Orientation
3. Copy Position
4. Copy X
5. Copy Y
6. Copy Z

Die Vorgehensweise beim Kopieren von Tag-Punkten ist stets die Selbe:

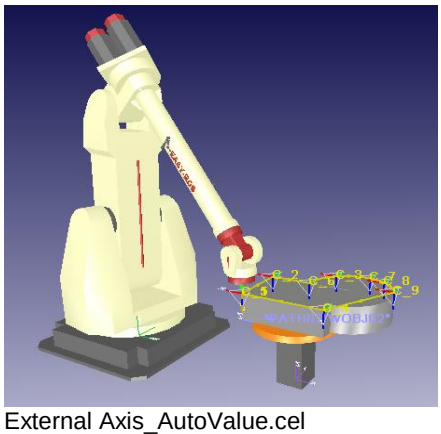
1. Öffnen Sie das Tag-Window mit einem Doppelklick auf 

2. Sie wählen zunächst aus, welche Tag-Punkte die Position und/oder Orientierung des „Referenz-Tag-Punktes“ übernehmen soll. Dabei ist die Reihenfolge der Tag-Punkt-Auswahl entscheidend. Der letzte in der Liste angewählte Tag-Punkt wird zum „**Referenz-Tag-Punkt**“.
3. Über *Manipulate Tags > Copy (Frame; Orientation; Pos; X; Y; Z)* können die ausgewählten Tag-Punkte dem Referenz-Tag-Punkt entsprechend angeglichen werden.
4. Kontrollieren und bestätigen Sie im folgenden Dialog Ihre Eingabe. Mit einem Klick auf „OK“ werden Eingaben übernommen.

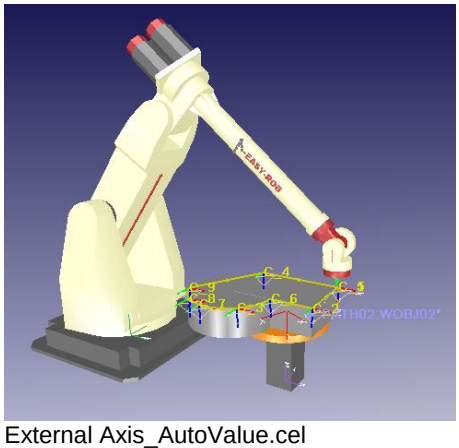
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Automatische Berechnung der externen Achswerte in Abhängigkeit der Pfadlänge

Die Werte für externe Achsen sind in den Tag Punkten abgelegt und müssen entsprechend der Aufgabe so bestimmt werden, dass z.B. die Positionen am Bauteil erreichbar sind. Mit der neuen Funktion „Axis-Value-Auto“ werden die externen Achswerte in Abhängigkeit der Pfadlänge automatisch bestimmt. Im Beispiel „External Axis_AutoValue.cel“ wird der Drehtisch durch die externe Achse 7 des Roboters gesteuert, siehe Bild



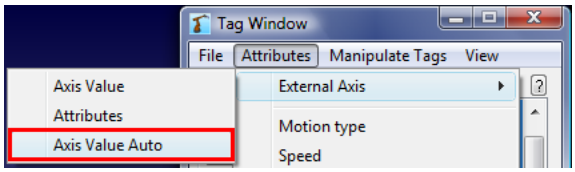
External Axis_AutoValue.cel



External Axis_AutoValue.cel

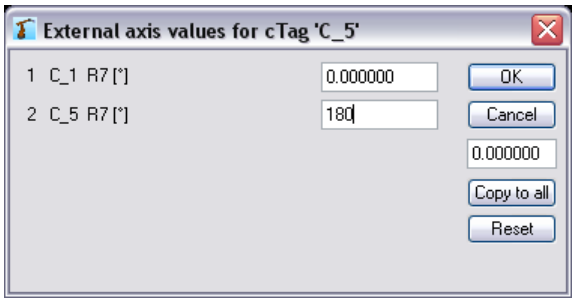
Über die Funktion **„Axis Value Auto“** kann eine gleichmäßige Drehbewegung des Tisches auf einen Pfadabschnitt eingestellt werden.

Die Funktion finden Sie im Tag-Window Menu Attributes > External Axis > Axis Value Auto



Wählen Sie die gewünschten Tag-Punkte bzw. den gewünschten Pfad im Tag Window aus und klicken Sie auf Attributes > External Axis > Axis Value Auto und geben den Start- und Endwert an.

Die angegebene Drehung wird nun gleichmäßig (in Abhängigkeit der Tag-Abstände, die von der Verfahrart abhängen) auf den gesamten ausgewählten Pfadabschnitt verteilt.

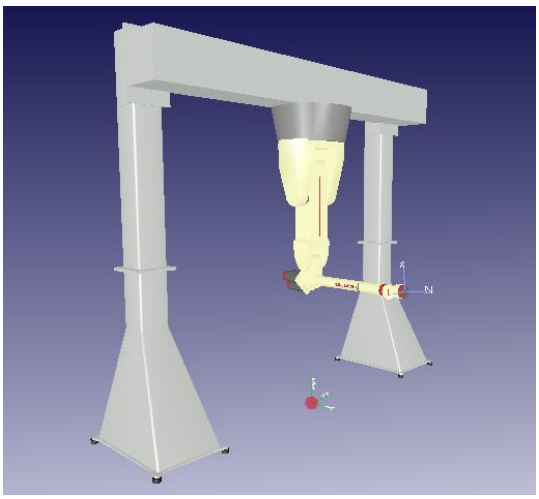


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Neuer Kinematik Typ: JET Roboter (Portalroboter)

JET Robotern mit der ID 127 (bzw. 128 für A2A3 Kopplung) mit der seriellen Struktur TyRyy:Rxyx oder TyRyy:Rzyz mit 4 Konfigurationen werden unterstützt und können so einfach auch vom Kunden erstellt werden. (z.B. KR 30 JET, KR 60 JET)

Zwei- und Drei-achsige Gantry-Kinematiken (ID 133) lassen sich in den Kombinationen Txyz, Tyxz, Tzxy, Tzyx, Txzy, Tyzx bzw. Txz, Tyz, Tzx, Tzy, Txy, Tyx erstellen



er431-jet.rob

Ein Beispiel des KR 30 JET

Auf der folgenden Seite finden Sie eine Liste mit Roboterkinematiken, die von EASY-ROB unterstützt werden.

Inverse Kinematics ID

Die „Inverse Kinematics ID“ legt die mathematische Lösung für die Rückwärts-Transformation (inverse kinematics) und Vorwärtstransformation für jeden Roboter fest. Für verschiedenste Kinematiken, wie beispielsweise 3- oder 5 Achs Portale, Knickarmroboter, Scaras, etc., bietet EASY-ROB™ direkte Lösungen an.

Zu jeder Kin-ID gibt es eine Sub-ID (default 0)

Die Kin-ID und die Sub-ID können im Robotics Menu:
cRobot Kinematics -> Kinematics Data -> Inverse Kinematics ID -> Special Inverse Kinematics
editiert werden.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Kin-ID Tabelle

Kin-ID	Name	Sub ID	Kin_Type	Kommentar
0				keine inverse Kinematik vorhanden
1	DLL #1			benutzerdefinierte Kinematik in „er_kin.dll“ #1
2	DLL #2			benutzerdefinierte Kinematik in „er_kin.dll“ #2
3	DLL #3			benutzerdefinierte Kinematik in „er_kin.dll“ #3
4	DLL #4			benutzerdefinierte Kinematik in „er_kin.dll“ #4
5	DLL #5			benutzerdefinierte Kinematik in „er_kin.dll“ #5
6	DLL #6			benutzerdefinierte Kinematik in „er_kin.dll“ #6
7	DLL #7			benutzerdefinierte Kinematik in „er_kin.dll“ #7
8	DLL #8			benutzerdefinierte Kinematik in „er_kin.dll“ #8
9	DLL #9			benutzerdefinierte Kinematik in „er_kin.dll“ #9
10	DLL #10			benutzerdefinierte Kinematik in „er_kin.dll“ #10
11	DLL #11			benutzerdefinierte Kinematik in „er_kin.dll“ #11
12	DLL #12			benutzerdefinierte Kinematik in „er_kin.dll“ #12
13-99	User Inverse Kinematics			benutzerdefinierte Kinematik in „er_kin.dll“ #13-99
100	NumSol	0	beliebig	Numerische Lösung Kinematik >= 6 Achsen Weitere Parameter: Tolerances, Joint Weight, Mask Vector
100	NumSol	1	beliebig	Numerische Lösung Kinematik mit weniger als 6 Achsen (Übereinstimmung Approach-Achse)
110	Knickarm		RzRyy:Rxyx , RzRyy:Rzyz	und Tracking-Achse „Standard RRR:RRR on Y-Track“
111	Knickarm		RzRyy:Rxyx , RzRyy:Rzyz	wie 110, mit „Backlink“ bzw. „A2A3 Kopplung“ „Back Link RRR:RRR on Y-Track“
116	Knickarm			wie 110, Lösung w.r.t Robot Base
117	Knickarm			wie 111, Lösung w.r.t Robot Base
122	Güdel	0, 10, 11, 12, 13		RoboFlex (Jet Roboter) xyz- Gantry xyz:Rz Gantry xz or yz Gantry xyz:Rz, yxz:Rz Gantry

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Kin-ID Tabelle

Kin-ID	Name	Sub ID	Kin_Type	Kommentar
120	b+m			T1 Lackierroboter
123	Denso	0, 1		Standard RRR:RRR on Y-Track Scara 4 axis RzRzTzRz
124	Mitsubishi	0, 1		Standard RRR:RRR on Y-Track Scara 4 axis RzRzTzRz
125	Eisenmann	0,1,2, 10		vrbh6,vrbc6,vrbl5, E-Shuttle
126	Adept	0, 1	0	Standard RRR:RRR on Y-Track Scara 4 axis RzRzTzRz
127	Jet Robot		TyRyy:Rxyx, TyRyy:Rzyz	
128	Jet Robot			wie 127, mit „Backlink“ bzw. „A2A3 Kopplung“
131	SCARA	0	RzRzTzRz	
133	Gantry 2 Achsen	13,23	Txz,Tyz, Tzx, Tzy, Txy, Tyx	2 Achs Portal
	Gantry 3 Achsen	123,0	Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx	3 Achs Portal
	Gantry 1 Achse	1,2,3	Tx, Ty, Tz	1 Achs Portal, Conveyor
134	Gantry 2+1 Achsen	13, 23	Txz,Tyz, Tzx, Tzy, Txy, Tyx, Rz	2 Achs Portal + Rz Drehachse
	Gantry 3+1 Achsen	123, 0	Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, Rz	3 Achs Portal + Rz Drehachse
135	Gantry, 3+2 Achsen		Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, CA=Rzx	5-Achs Portal mit C- und A-Achse
136	Gantry 6 Achsen		Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, Rzxz,Rzyz	6-Achs Portal mit Rzxz oder Rzyz Handgelenk
114	Abb			Optional
118	Motoman			Optional
115	Staubli			Optional
112	Kuka			Optional
113	Fanuc			Optional
132	Tricept			Optional
137	PKM			Optional, Delta-Kinematik, FlexPicker

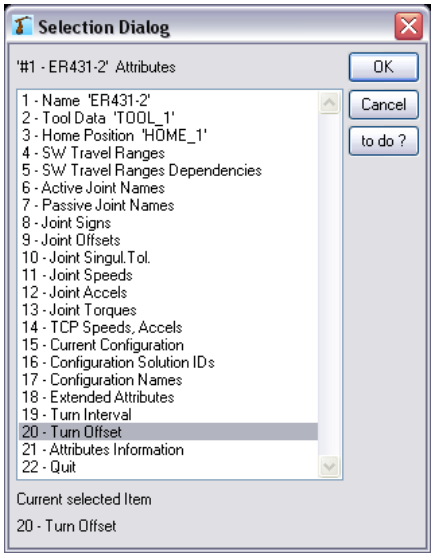
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Neue Roboter-Attribute

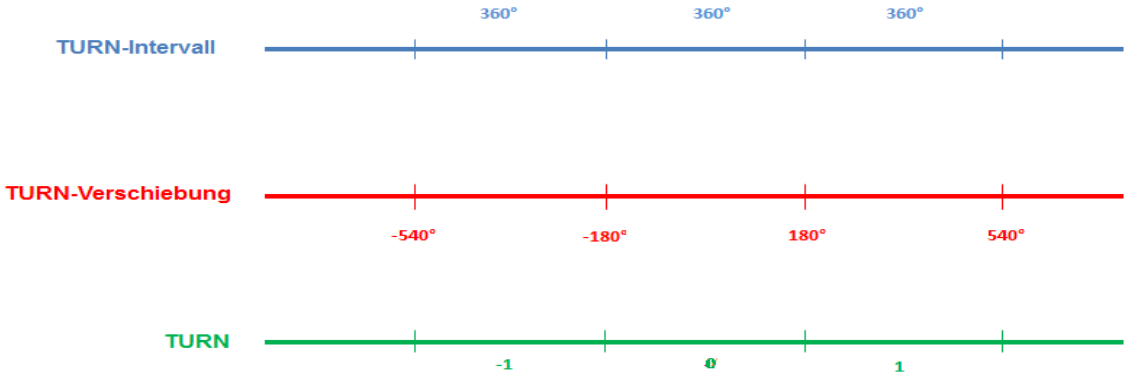
Für aktive- und passive Achsen von Robotern, Kinematiken bzw. Geräten (Devices) können Namen vergeben werden, z.B. Hubachse, Querantrieb oder Achse_1 anstatt Joint_1.
Auch lassen sich Namen für die Konfigurationen vergeben, z.B. für Kuka S'B010 oder Fanuc NUT anstatt Config_1.
Neben dem Turn-Intervall können auch Turn-Offsets für jede Achse definiert werden, was für den Abgleich mit der realen Roboter-Steuerung unablässig ist

Turn-Offset

Öffnet Sie dazu das Kinematics Window und klicken auf Attributes



Die nachfolgende Grafik soll Turn-Offset näher erläutern



Bei einem Turn-Intervall von 360° und einem Turn-Offset von -180° ergibt sich der Turn = 0 wenn q in]-180,180], = 1 in]180,540] und = -1 im Bereich]-540,-180].

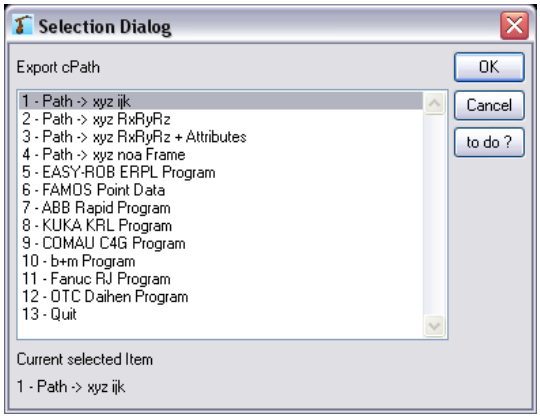
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

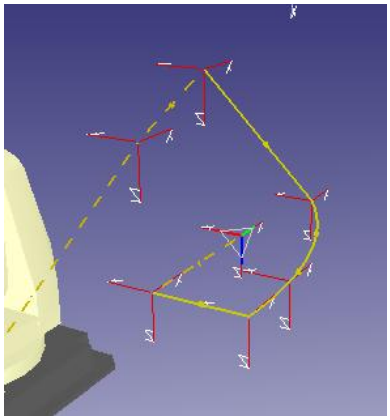
Export von Pfaden in native Roboterprogramme

Der generische Export von Pfaden in native Roboterprogramme ist für die Steuerungstypen von ABB, KUKA, COMAU, Fanuc, b+m und OTC erweitert worden. Weitere werden je nach Kundenwunsch eingebunden

Als Beispiel soll die Zelle „Path-Export.cel“ dienen. Der Pfad hat 7 Tags, die in native Roboterprogramme exportiert wurden. Öffnen Sie dazu das Tag Window und wählen im Menu File -> Save -> Export cPath

Export Path erlaubt es mit EASY-ROB™ erzeugte Bahnen syntaktisch korrekt in Roboterprogramme zu übernehmen. Manche nennen das sogar OLP?!? Wir empfehlen an dieser Stelle dann doch eher Famos robotic®, ein richtiges Offline Programmierwerkzeug für Roboter mit Prozess-Technologie.




Pfad mit 7 Tags

Folgend einige Export-Beispiele

Export Path ABB

Ausgabe-Datei Path-Export-ABB.prg

```
MODULE MY_PROG
! ---------------------------------------------
! Path Export
! EASY-ROB 3D Robot Simulation Tool
! Copyright (c) 1996-2012
!
! cRobot ER431-2
! ---------------------------------------------
! ---------------------------------------------------------
! Pathnumber: PATH01
! ---------------------------------------------------------
PROC ErProg001()
 MoveJ T_1,vProcess,fine,TOOL_1\WObj:=WOBJ01;
 MoveJ T_2,vProcess,fine,TOOL_1\WObj:=WOBJ01;
 MoveL T_3,vProcess,fine,TOOL_1\WObj:=WOBJ01;
 MoveC T_4,T_5,vProcess,fine,TOOL_1\WObj:=WOBJ01;
 MoveL T_6,vProcess,fine,TOOL_1\WObj:=WOBJ01;
 MoveJ T_7,vProcess,fine,TOOL_1\WObj:=WOBJ01;
ENDPROC
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Export Path Kuka

Ausgabe-Datei Path-Export-Kuka.dat, Path-Export-Kuka.src

```
&COMMENT EASY-ROB Data (Build: 1)
DEFDAT MY_PROG PUBLIC
; ----------------------------------------------
; Path Export
; EASY-ROB 3D Robot Simulation Tool
; Copyright (c) 1996-2012
; cRobot ER431-2
; ----------------------------------------------
; BASE
FRAME WOBJ01 = {X -0.00,Y -0.00, Z -0.00, A 0.00,
B 0.00, C 0.00}
; TOOL
FRAME TOOL_1 = {X 0.00,Y 0.00, Z 150.00, A 0.00,
B 0.00, C 0.00}
; Targets of Path: PATH01
FRAME T_1 = {X 1599.89,Y -626.80, Z 1685.67, A
0.00, B 180.00, C 0.00}
FRAME T_2 = {X 1599.90,Y -215.80, Z 1853.09, A
0.00, B 180.00, C -0.00}
FRAME T_3 = {X 1599.90,Y 780.00, Z 1066.07, A
0.00, B 180.00, C 0.00}
FRAME T_4 = {X 1903.70,Y -114.85, Z 1066.06, A
0.00, B 180.00, C 0.00}
FRAME T_5 = {X 1903.71,Y -445.23, Z 1066.05, A
0.00, B 180.00, C 0.00}
FRAME T_6 = {X 1505.09,Y -445.23, Z 1066.05, A
0.00, B 180.00, C 0.00}
FRAME T_7 = {X 1505.09,Y 242.77, Z 1053.92, A
0.00, B 180.00, C 0.00}
```

```
&COMMENT EASY-ROB Program (Build: 1)
DEF Path_Export_Kuka()

BAS (#INITMOV,0 ) ;Initialisierungen

 $IPO_MODE = #BASE
 $ORI_TYPE = #VAR
 $CIRC_TYPE = #BASE
 $TOOL=TOOL_1
 $BASE=WOBJ01
 $VEL.CP = 0.255
 $APO.CDIS = 0.1
 $APO.CORI = 0.100
 $APO.CVEL = 100
 BAS (#VEL_PTP,26)
 PTP T_1
 PTP T_2
 LIN T_3
 CIRC T_4,T_5
 LIN T_6
 PTP T_7
END
```

Export Path Comau

Ausgabe-Datei Path-Export-Comau.txt

```
-------------------------------------------------------------------------------
-- COMAU C4G Program Export
-- EASY-ROB 3D Robot Simulation Tool
-- Copyright (c) 2012
--
-- FILE:    Path-Export-Comau.PDL
-------------------------------------------------------------------------------

PROGRAM Path-Export-Comau
CONST
VAR -- global data

BEGIN -- main program

  $ORNT_TYPE := RS_WORLD
  $MOVE_TYPE := JOINT
  $CNFG_CARE := FALSE
  $TURN_CARE := FALSE
  $SING_CARE := FALSE
  $TERM_TYPE := NOSETTLE
  $LIN_SPD := 0.2
  $ROT_SPD := 3.1415926
  $SPD_OPT := SPD_LIN
  --
  $BASE := POS(-0.0000,-0.0000,-0.0000,0.000,0.000,0.000,'')
  $TOOL := POS(0.0000,0.0000,150.0000,0.000,0.000,0.000,'')

  -- Execution
  MOVE JOINT TO POS(1599.8870,-
626.7960,1685.6689,0.000,180.000,0.000,'')
  MOVE JOINT TO POS(1599.8960,-
215.7960,1853.0910,90.000,180.000,90.000,'')
  MOVE LINEAR TO POS(1599.8991,779.9990,1066.0700,-
90.000,180.000,-90.000,'')
  MOVE CIRCULAR TO POS(1903.7080,-445.2330,1066.0480,-
90.000,180.000,-90.000,'')  VIA POS(1903.7000,-
114.8540,1066.0599,-90.000,180.000,-90.000,'')
  MOVE LINEAR TO POS(1505.0910,-445.2340,1066.0460,-
90.000,180.000,-90.000,'')
  MOVE JOINT TO POS(1505.0920,242.7700,1053.9160,-
90.000,180.000,-90.000,'')
END Path-Export-Comau
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Export Path Fanuc

Ausgabe-Datei Path-Export-Fanuc.ls

```
/PROG  Path-Export-Fanuc    PROCESS
/ATTR
OWNER                = MNEDITOR;
COMMENT                      = "Path-Export-Fanuc";
PROG_SIZE            = 0;
CREATE               = ;
MODIFIED             = ;
FILE_NAME            = Path-Export-Fanuc;
VERSION              = 0;
LINE_COUNT           = 0;
MEMORY_SIZE          = 0;
PROTECT                      = READ_WRITE;
TCD:  STACK_SIZE             = 0,
      TASK_PRIORITY          = 50,
      TIME_SLICE     = 0,
      BUSY_LAMP_OFF          = 0,
      ABORT_REQUEST          = 0,
      PAUSE_REQUEST          = 0;
DEFAULT_GROUP= 1,*,*,*,*;
CONTROL_CODE  = 00000000 00000000;
/APPL
/MN
   1:  !UFrame[0] = X = -0.000 mm, Y = -0.000 mm, Z = -0.000
mm, W = 0.000 deg, P = 0.000 deg, R = 0.000 deg;
   2:  !UTool[0]  = X = 0.000 mm, Y = 0.000 mm, Z = 150.000
mm, W = 0.000 deg, P = 0.000 deg, R = 0.000 deg;
   3:  !---Program Begin;
   4:J P[1] 50% FINE;
   5:J P[2] 50% FINE;
   6:L P[3] 800mm/sec FINE;
   7:C P[4] P[5] 800mm/sec FINE;
   8:L P[6] 800mm/sec FINE;
   9:J P[7] 50% FINE;
  10:  !---Program End;
/POS
P[1]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1599.887 mm, Y = -626.796 mm, Z = 1685.669 mm,
     W = 180.000 deg, P = 0.000 deg, R = 180.000 deg
};
P[2]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1599.896 mm, Y = -215.796 mm, Z = 1853.091 mm,
     W = 180.000 deg, P = 0.000 deg, R = -180.000 deg
};
P[3]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1599.899 mm, Y = 779.999 mm, Z = 1066.070 mm,
     W = -180.000 deg, P = 0.000 deg, R = -180.000 deg
};
P[4]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1903.700 mm, Y = -114.854 mm, Z = 1066.060 mm,
     W = -180.000 deg, P = 0.000 deg, R = -180.000 deg
};
P[5]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1903.708 mm, Y = -445.233 mm, Z = 1066.048 mm,
     W = -180.000 deg, P = 0.000 deg, R = -180.000 deg
};
P[6]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1505.091 mm, Y = -445.234 mm, Z = 1066.046 mm,
     W = -180.000 deg, P = 0.000 deg, R = -180.000 deg
};
P[7]{
  GP1:
     UF : 0, UT : 1,      CONFIG : 'F U T, 0, 0, 0',
     X = 1505.092 mm, Y = 242.770 mm, Z = 1053.916 mm,
     W = -180.000 deg, P = 0.000 deg, R = -180.000 deg
};
/END
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Kollision

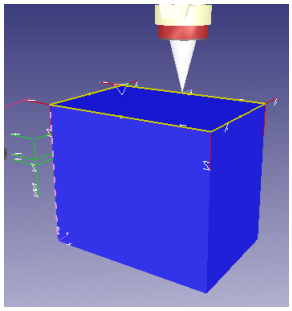
Der neue Kollisions-Algorithmus „PQP“ erlaubt es Toleranzen zu definieren. Demnach wird Kollision angezeigt wenn zwei Körper sich auf einen minimalen Abstand nähern.

Dieser Toleranzwert kann nun für **jeden einzelnen Körper** individuell vorgegeben werden.

Geometriespezifischer Toleranzwert

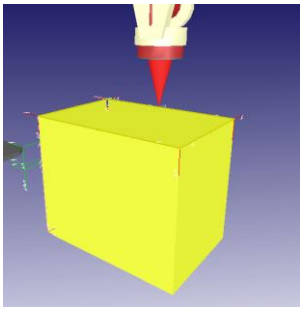
Beispiel: Collision-Tolerance.cel

Bei diesem Beispiel werden die Collisions-Tolerances mit ERPL –Behlen im Programm variiert.



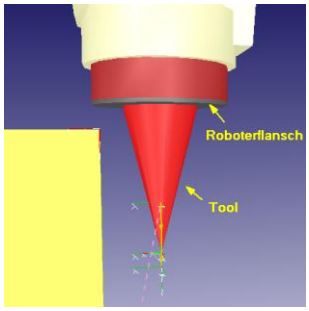
Collision-Tolerance.cel , zuerst ohne Kollision

In diesem Beispiel wird zuerst entlang des Cubes mit einer Collision-Tolerance von 0 mm gefahren, so dass es keine Kollision gibt.



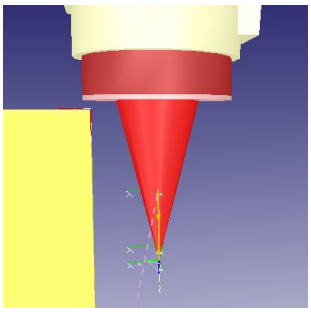
Collision-Tolerance.cel, mit Kollision

Im zweiten Fall wird eine globale Kollision von 5 mm eingestellt, so dass nun Kollision auftritt.



Für die Tool-Geometrie ist eine Kollisionstoleranz von 50 mm und für den Roboterflansch eine Kollisionstoleranz von 20 mm definiert.

Ergebnis:
Tool kollidiert mit Cube, da die Distanz < 50 mm ist
Flansch kollidiert nicht mit Cube, da die Distanz > 20 mm ist.



Robot fährt noch ein Stück tiefer

Ergebnis:
Tool kollidiert weiterhin mit Cube
Flansch kollidiert nun auch mit dem mit Cube, da die Distanz < 20 mm ist.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Neue ERPL und ERCL Befehle

MSG Text

“Text“ wird im Program Window angezeigt

NATIVE Native-Text

Der native Befehl kann direkt in das Post-Processor API verwendet werden.
Er kann keinen Einfluss auf die Simulation.

ERC TURN_INTERVAL Ax1 ...Axn [deg]

TURN -Intervalle für jede Achse Ax1…Axn, im Bereich zwischen [0°,∞°]

ERC TURN_OFFSET Ax1 ...Axn [deg]

TURN - Offset für jede Achse Ax1…Axn, im Bereich zwischen]-∞°, ∞°[

ERC GRAB_TO DEVICE devname targetdevname

Das Gerät mit dem Namen ’devname’ wird vom Gerät mit dem Namen ‘targetdevname’ gegriffen.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

API Application Program Interface, Methoden Klasse ER_CAPI

Für individuelle Produktanpassungen und spezielle Lösungen sind viele neue API Funktionen hinzugekommen. Diese ermöglichen es beispielsweise EASY-ROB™ aus einer eigenen Applikation anzusteuern bzw. bidirektional Daten auszutauschen. Die Methoden-Klasse ER_CAPI dient als Schnittstelle für die EASY-ROB™ Multi-Program und EASY-ROB™ DLL Version sowie für die Erweiterungen API-INV, API-IPO, API-DYN, API-UserDLL, API-PostProc und API-Sensors.

Die exportierte Klasse ER_CAPI strukturiert sämtliche EASY-ROB™ API-Funktionen und vereinfacht die Verwendung. Die Klasse ER_CAPI ist eine reine Methoden-Klasse, die in der Header-Datei „./er_dvlp/**er_capi.h**“ „./er_dvlp/**er_capi_types.h**“und definiert sind. Sämtliche Methoden sind Standard ANSI C. „Alte“ ANSI C Funktionen, die in den Header-Dateien „./er_dvlp/**er_dvlp.h**“ und „./er_dvlp/**er_dvlp_ext.h**“ definiert sind, sind aus Kompatibilitätsgründen weiterhin verfügbar.

AUX_UPDATE_IDX_SET_FOCUS
ER_CAPI_SYS_MATHEMATICS::circ_center_point() ber. Zus. Den Winkel bis zum Via-Pkt.

ROB_KIN

- **void **ER_CAPI_ROB_DYN::inq_kin_usr_ptr (void)**

```
// access user pointer for user kinematics
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() below
```

ROB_DYN

- **void **ER_CAPI_ROB_DYN::inq_dyn_cntrl_usr_ptr(void)**

```
// access user pointer for dynamics controller
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() below
```

- **void **ER_CAPI_ROB_DYN::inq_dyn_model_usr_ptr(void)**

```
// access user pointer for dynamics model
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() below
```

- **void **ER_CAPI_ROB_DYN::inq_status_output_usr_ptr(void)**

```
// access user pointer for status ouput
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() below
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

MOP

- **void **ER_CAPI_MOP::inq_ipo_jnt_usr_ptr(void)**

```
// access user pointer for ipo joint
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() below
```

- **void **ER_CAPI_MOP::inq_ipo_cp_usr_ptr(void)**

```
// access user pointer for ipo cp
// Example how to access and use user pointer

static ER_CAPI_MOP er_mop; // Method ER_CAPI_MOP, see file er_capi.h
IPO_USR_CP *ipo_ptr; // pointer to your user defined structure
void **ipo_cp_usr_ptr = er_mop.inq_ipo_cp_usr_ptr(); // access user pointer
if (*ipo_cp_usr_ptr==NULL)                  // check if allocation is done
{
    *ipo_cp_usr_ptr = (void *)malloc(sizeof(IPO_USR_CP)); // alloc once
    ipo_ptr = (IPO_USR_CP *)*ipo_cp_usr_ptr;
    if (ipo_ptr==NULL) {
        _info_line_msg(1,"Error, malloc IPO_USR_CP");
        return 1;                           // return error
    }
    //intialize values
    ipo_ptr->my_value=0;                    // example
}
// continue with calculation
ipo_ptr = (IPO_USR_CP *)*ipo_cp_usr_ptr;// access address
return 0;                                   // return ok
```

- **void **ER_CAPI_MOP::inq_ipo_circ_usr_ptr(void)**

```
// access user pointer for ipo circ
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() above
```

MOP_PATH

- **int *ER_CAPI_MOP_PATH::inq_ipo_path_path_via_motion_idx(void)**

```
// 0-no tag via motion, >0 Index of cPath containing the TargetTagVia
//
```

- **int *ER_CAPI_MOP_PATH::inq_ipo_path_tag_via_motion_idx(void))**

```
// 0-no tag via motion, >0 Index of TargetTagVia
//
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

SIM_ERPL

- **void **ER_CAPI_SIM_ERPL::inq_pp_usr_ptr()**

```
// access user pointer for post_processor in er_post.dll
// see example ER_CAPI_MOP::inq_ipo_cp_usr_ptr() above
```

TARGETS_TAG

- **frame *ER_CAPI_TARGETS_TAG::inq_rob_tag_T_cRBase_idx(int idx)**

```
// tag_crobot data are temporarily and calculated when the
// cRobot moves to a Tag
// tag_crobot, tmp Tcp location w.r.t to cRobot Base
//
```

- **float *ER_CAPI_TARGETS_TAG::inq_rob_tag_q_cR_idx(int idx**

```
// tag_crobot data are temporarily and calculated when the
// cRobot moves to a Tag
// tag_crobot, tmp joint location for cRobot
//
```

- **ER_UID *ER_CAPI_TARGETS_TAG::inq_rob_tag_uid_cR_idx(int idx**

```
// tag_crobot data are temporarily and calculated when the
// cRobot moves to a Tag
// tag_crobot, tmp uid for cRobot
//
```

TARGETS_PATH

- **static float *ER_CAPI_TARGETS_PATH::inq_Get_c_path_len()**

```
// get lengths of complete path or selected path
//
```

- **static float *ER_CAPI_TARGETS_PATH::inq_Get_c_path_angle_len()**

```
// get angle lengths of complete path or selected path
//
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

- **static float ER_CAPI_TARGETS_PATH::inq_Calc_path_length(int path_idx, int tag_idx_strt, int tag_idx_end, int ret_angle)**

```
// calculates length/angle of path
// IN: path_idx, tag_idx_strt, tag_idx_end
// IN: reg_angle, 0 - calc length, 1 - calc angle
// Return: length
//
```

CAD_IO

- **void *ER_CAPI_CAD_IO::inq_body_obj_attributes_handle(void *body_handle,int i_obj**

```
// return  VRML_OBJ_ATTRIBUTES *obj_handle_attributes
// IN: _BODYS *body_handle, int i_obj
//
```

- **float *ER_CAPI_CAD_IO::inq_body_obj_color_rgba(void *obj_handle_attributes)**

```
// IN: VRML_OBJ_ATTRIBUTES *obj_handle_attributes
// return rgba color
//
```

- **int *ER_CAPI_CAD_IO::inq_body_obj_ptrcolor(void *obj_handle_attributes)**

```
// IN: VRML_OBJ_ATTRIBUTES *obj_handle_attributes
// return color pointer
//
```

SYS_MATHEMATICS

- **int ER_CAPI_SYS_MATHEMATICS::circ_center_point(float *p1,float *p2, float *p3,frame *pTc, float *radius,float *phi, float *dphi=NULL)**

```
// circle calculation from
// IN p1 over p2 to p3.
// OUT pTc circle center, radius,
//     phi is from p1 to p3,
//     dphi is from p1 to p2.
// Return 0 - ok, 1 - error
//
```


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Sonstige Erweiterungen

- Teach-Window mit neuem Dialog mit sämtlichen Bewegungsbefehlen
- Device Manager Dialog nun skalierbar
- AVI-Recorder mit weiteren Auflösungen
- 3D Space Mouse mit einstellbarer Empfindlichkeit und Schwellwert in Umgebungsdatei
- Neue Parser-Funktionen mit Zugriff auf kinematische Roboterlängen und mehr
- Erweiterter Status Output für die Ausgabe von z.B. Achswerten in jedem Simulationsschritt

Konfiguration der Space Mouse über die Umgebungsdatei

Bild: 3DConnexion

Über die Umgebungsdatei (Alt+Shift+E) „easy-rob.env“ können grundlegende Funktionen der Space Mouse angepasst werden:

```
S3DM_MENU 1
! Scales Space Mouse sensitivity
S3DM_SPEED 1.000000
! Scales Space Mouse threshold
S3DM_THRESHOLD 1.000000
```

Bezeichnung	Wert	Funktion
S3DM_MENU	0;1	An-/Abschalten des Space Mouse Menüs
S3DM_SPEED	1	Empfindlichkeit der Maus einstellen
S3DM_THRESHOLD	1	Konfiguriert den „Ansprechwert“ der Space Mouse. Mit dem von Ihnen gesetzten Wert bestimmen Sie, wie weit die Maus bewegt werden muss, bis sie anspricht.

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Kontakt

EASY-ROB 3D Robot Simulation Tool

Stefan Anton
Hans - Thoma - Str. 26a, 60596 Frankfurt/Main, Germany
Tel. +49 (0) 69 677 24 287
Fax. +49 (0) 69 677 24 320

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

EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com