

Update

EASY-ROB™ V5.006

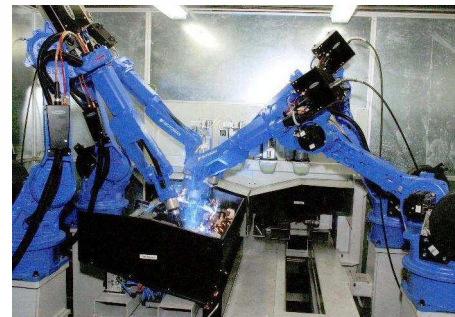
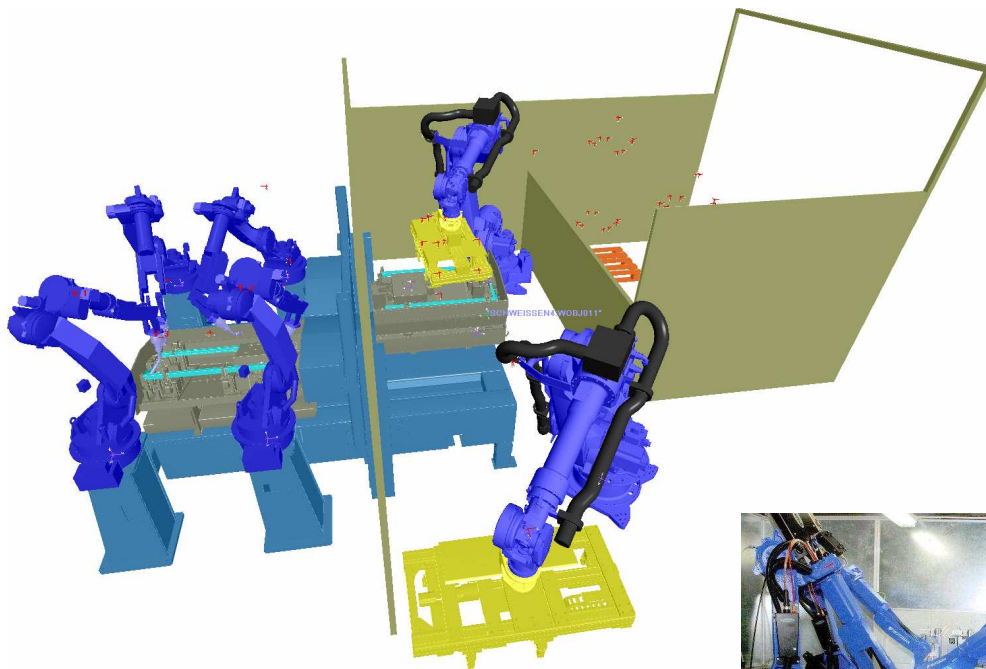


Bild: WESOBA GmbH, Schwarzenberg

Juni 2009

Version 1.01

EASY-ROB™

Inhaltsverzeichnis

Einleitung	5
EASY-ROB™ Viewer	7
CAD Daten-Import.....	8
Modellierung von Kinematiken mit Synchronisation	9
Schritte zum Ändern der Synchronisationseigenschaften.....	10
Externer TCP	11
EASY-ROB™ Lizenz Manager als Dienst	12
Genauere Taktzeitberechnungen	13
AUTO_ACCEL.....	13
Vergleich: Auto_Accel OFF/ON bei v=50% und 100%	14
ACCSET	15
Vergleich: ACCSET bei 20%, 50%, 80% und 100%	16
Messen.....	17
Abstände zwischen Devices messen	17
Kreismittelpunkt messen	17
ERPL-Befehle	18
ERCL-Befehle	19
Parser-Funktionen.....	19
API-Funktionen	20
USER_IO_DIALOG	20
SIM_ERPL – Simulation ERPL	20
DEVICES.....	21
ROB_KIN – Robot Kinematics	22
MOP_PATH – Motion Planner Path	23
Kontakt	25
Eigene Notizen.....	26

EASY-ROB™ V5.006

Einleitung

Das neue EASY-ROB™ Update Version 5.006 stellt wieder eine Vielzahl an Neuerungen und Verbesserungen zur Verfügung. Folgendes Dokument vermittelt Ihnen einen ersten Überblick über die wichtigsten Änderungen. Eine detaillierte Beschreibung finden Sie wie immer in den Bedienungshinweisen.

- **EASY-ROB™ Viewer**
Mit dem neuen kostenlosen EASY-ROB™ Viewer lassen sich vorhandene Arbeitszellen simulieren. Programmspezifische Funktionen wie unterschiedliche Perspektiven, Start, Stopp oder das Erzeugen von Video Dateien stehen zur Verfügung. Diese Anwendung wurde insbesondere für Vertriebsmitarbeiter als Präsentationsmittel erstellt.
- **CAD Daten-Import noch komfortabler**
Für die 3D Visualisierung in EASY-ROB™ ist der CAD Daten-Import von nahezu allen gängigen Formaten unverzichtbarer Bestandteil. Beim Import von Baugruppen oder Assemblies werden nun sofort Device-Files erzeugt, so dass sich die Komponenten individuell ausblenden oder gar löschen lassen.
- **Verbessert: Modellierung von Kinematiken mit Synchronisation**
Achsen einer Kinematik lassen sich nun beliebig und unabhängig voneinander definieren. Individuelle Greifer, Stanz- Umformmaschinen, etc. können einfacher erzeugt werden und externe Achsen werden besser in den Simulationslauf eingebunden. Die neue Möglichkeit Kinematiken miteinander zu synchronisieren stellt eine Vielzahl neuer Simulationsmöglichkeiten dar.
- **Externer TCP**
Werkstückführende Bewegungen werden nun besser unterstützt, insbesondere beim Erzeugen und Ändern von Tag-Punkten am Werkstück.
- **EASY-ROB™ Lizenz Manager als Dienst**
Der Lizenz Manager kann nun als Dienst installiert werden, was die Administration vereinfacht.
- **Genauere Taktzeitberechnungen**
Unter realen Bedingungen wurden Praxistests in Form von echten Messungen durchgeführt. Die Ergebnisse werden von der Bewegungsplanung vollständig berücksichtigt. Taktzeiten lassen sich nun noch genauer abschätzen.

Neben weiteren kleineren Änderungen sind neue ERCL- und ERPL-Befehle und API-Funktionen für die externe Ansteuerung der neuen Funktionen hinzugekommen. Das Update steht für alle Kunden mit einer gültigen Lizenz für EASY-ROB™ V5.0 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™ Viewer

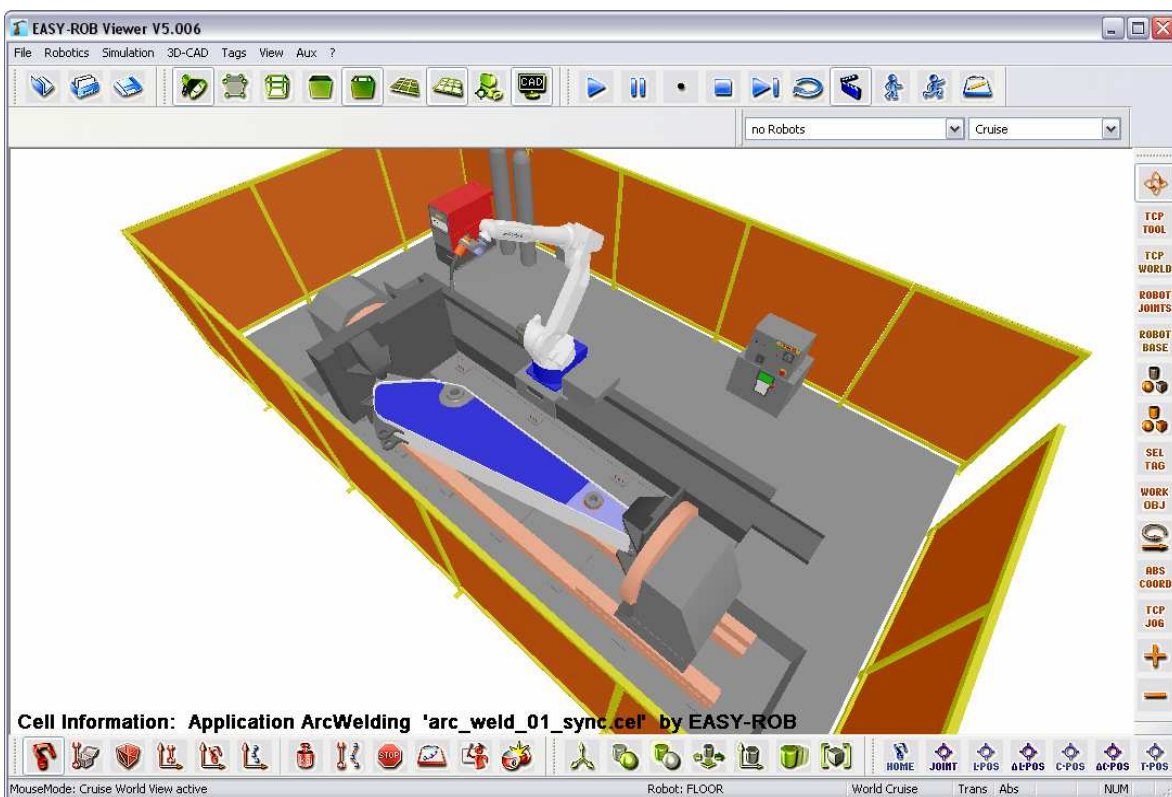
Mit dem neuen kostenlosen EASY-ROB™ Viewer lassen sich vorhandene Arbeitszellen simulieren. Nach dem Laden einer Arbeitszelle stehen sämtliche programmspezifische Funktionen wie z.B. unterschiedliche Perspektiven, Simulations-Start, -Stopp, -Pause oder beispielsweise das Erzeugen von Video Dateien und Screenshots zur Verfügung.

Diese Anwendung wurde insbesondere für Vertriebsmitarbeiter als Präsentationsmittel erstellt. Der EASY-ROB™ Viewer zählt zu den portablen Anwendungen und muss nicht installiert werden. So ist es möglich den Viewer direkt vom USB-Stick oder von der CD/DVD zu starten und die Projekte, die sich ebenfalls auf dem Datenträger befinden zu laden und zu simulieren.

Der EASY-ROB™ Viewer kann direkt von der aktuellen EASY-ROB™ CD aus dem Verzeichnis „.\Easy_Rob_Viewer“ kopiert werden. Alternativ lässt sich der Viewer durch das Ausführen des Setups „EASY-ROB Viewer Setup.exe“ auch installieren.

Voraussetzung für die sinnvolle Nutzung des EASY-ROB™ Viewers ist die Lizenzierung der Option **RunTime** in der EASY-ROB™ Version.

Mit der Option **RunTime** werden Ihre Arbeitszellen geschützt gespeichert und können so im Viewer geladen werden.

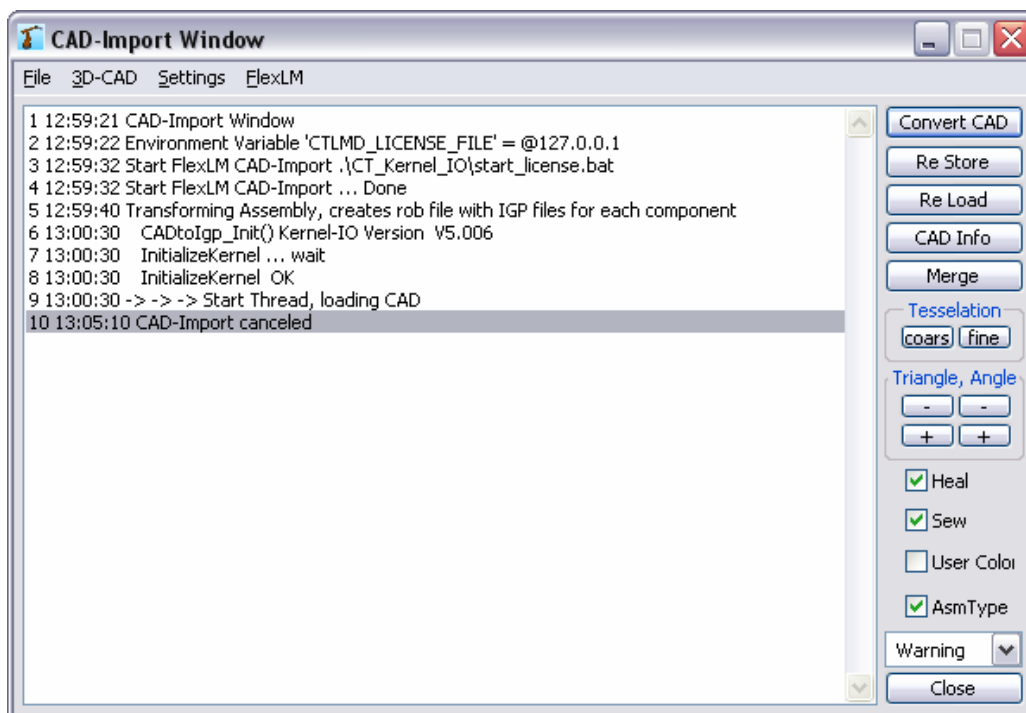


CAD Daten-Import

Mit dem leistungsfähigen CAD Daten-Import (3D_Evolution® API) können nahezu alle gängigen Formate importiert werden. Hierzu zählen zum einen die neutralen Formate wie STEP, IGES, VDA, VRML II und JT-Open als auch die nativen Formate wie beispielsweise CATIA V4 und CATIA V5 R19, Pro/E, UG, SolidWorks, Robface und weitere.

Baugruppen und Assemblies auflösen.

Wurde in der Vorgängerversion die gesamte Baugruppe oder das gesamte Assembly in eine einzige Geometriedatei (Igp-Datei) konvertiert, ist es nun möglich die einzelnen Komponenten jeweils als Igp-Datei zu konvertieren. Das gleichzeitig erzeugte Rob-File erlaubt es nach dem Laden ungewünschte oder nicht erforderliche Komponenten auszublenden oder zu löschen. Alternativ können einzelne Komponenten neu positioniert und kinematisiert bzw. bewegten Achsen zugeordnet werden. Das Erzeugen von Kinematiken vereinfacht sich damit erheblich.



CAD-Import Window: Menü „3D-CAD | Open CAD-Import Window“.

Aktivieren Sie die Check-Box AsmType, um für jede Komponente ein IGP file und ein Rob file zu erzeugen. Mit „Convert CAD“ wählen Sie die zu konvertierende CAD-Datei aus.

Ist die Check-Box „AsmType“ nicht aktiviert, wird aus der importierten CAD Datei ein IGP file erzeugt.

Zwei Beispiele befinden sich auf der EASY-ROB™ CD in den Verzeichnissen

```

..\Options\CAD-Import\Iges\AsmType\P-500 Top Machine.IGS" -> P-500TopMachine.rob
..\Options\CAD-Import\Step\AsmType\KR500-2_KR360-2.stp " -> KR500-2_KR360-2.rob
  
```

Modellierung von Kinematiken mit Synchronisation

Mit EASY-ROB™ lässt sich nicht nur das Bewegungsverhalten von Industrie-Robotern simulieren, sondern auch verschiedenste Kinematiken wie Greifer, Positionierer, Zuführ- und Fördereinheiten, Dreh-, Kipp-Schwenktische, Vorrichtungen und Sondermaschinen.

Um diesen Anforderungen gerecht zu werden, stehen weitere Möglichkeiten bei der Definition der Achsen zur Verfügung.

Achsen einer Kinematik lassen sich nun beliebig und unabhängig voneinander definieren und müssen nicht mehr zwangsläufig Teil der kinematischen Haupt-Kette sein.

Hinzu kommt die Möglichkeit Kinematiken miteinander zu synchronisieren. Hierbei werden die Achsen einer Kinematik mit der einer anderen gekoppelt, werden also gewissermaßen zum Slave einer Master-Kinematik.

Typische Anwendung

Robotersteuerungen steuern einerseits den Roboter mit 6 Achsen und zusätzlich die Peripherie, die externen Achsen, wie z.B. eine Lineareinheit und einen Positionierer. Mit dem Ziel diesen Sachverhalt gleichwertig in EASY-ROB™ nachzubilden, wird die Roboterkinematik um 2 weitere unabhängige Achsen 7 und 8 ergänzt. Werden Lineareinheit und Positionierer mit dem Roboter synchronisiert, lassen sich diese Geräte vollständig vom Roboter steuern. Das vereinfacht die Programmierung von externen Achsen erheblich. So können die externen Achswerte auch in den Tag-Punkten abgelegt werden. Wird solch ein Tag-Punkt angefahren bewegen sich Roboter zusammen mit Lineareinheit und Positionierer zeitlich synchronisiert. Die Bewegung der drei Kinematiken beginnt und endet gleichzeitig.

Features

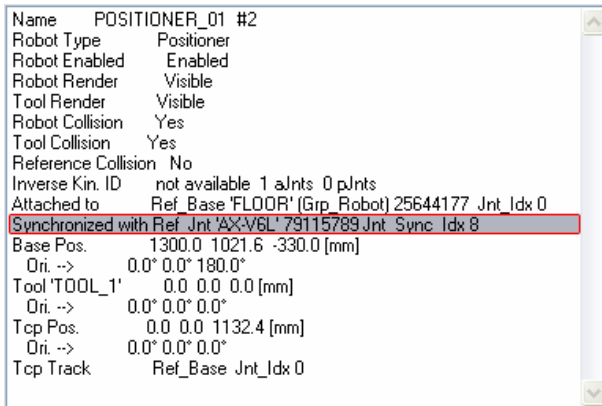
- Es können mehrere Kinematiken mit einer anderen Kinematik synchronisiert werden.
- Es können einzelne Achsen einer Kinematik synchronisiert oder eigenständig sein.
- Synchronisierte Kinematiken können ein eigenes Programm zur Ansteuerung haben.

Diese neuen Möglichkeiten erlauben es eine Vielzahl von Szenarien mit EASY-ROB™ simulationstechnisch zu untersuchen.

Beispiel: „.\ApplicationLib\ArcWelding\arc_weld_01_sync.cel“

In diesem Beispiel ist der Drehpositionierer „POSITIONER_01“ mit der 8. Achse des Roboters „AX-V6L“ synchronisiert. Die Einstellung kann im Kinematics Window durch Doppelklick auf die Zeile „Synchronized with Ref Jnt , AX-V6L' ... “ geändert werden.

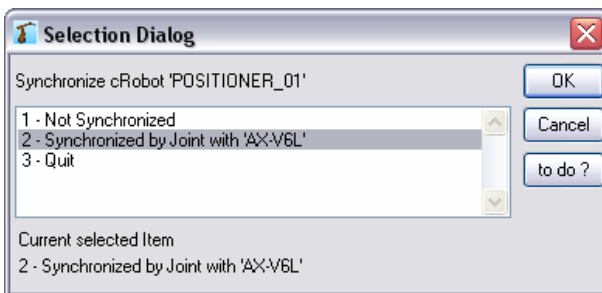
Schritte zum Ändern der Synchronisationseigenschaften



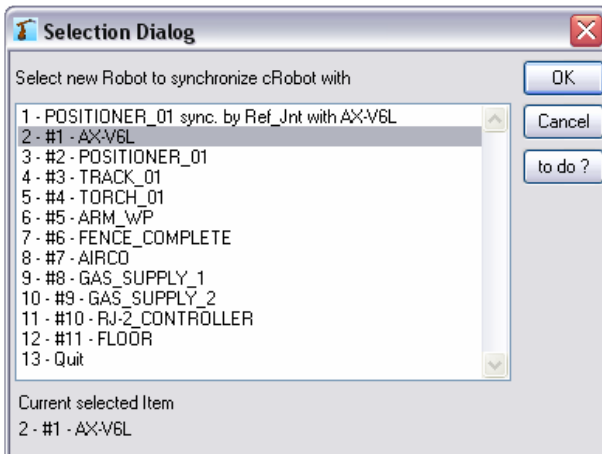
Beispiel: „\ApplicationLib\ArcWelding\arc_weld_01_sync.cel“

In diesem Beispiel ist der Drehpositionierer „POSITIONER_01“ mit der 8. Achse des Roboters „AX-V6L“ synchronisiert. Die Einstellung kann im Kinematics Window durch Doppelklick auf die Zeile „Synchronized with Ref Jnt ‚AX-V6L‘ ... “ geändert werden.

1. Doppelklick auf „Synchronized with...“ zur Änderung der Eigenschaften



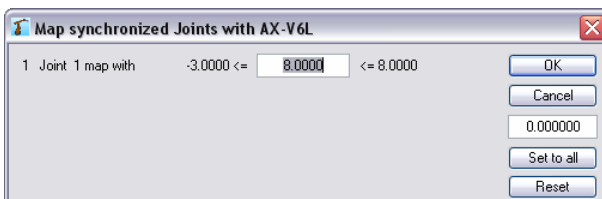
2. Auswahl:
- Not Synchronized
- Synchronized by Joint with ‚AX-V6L‘



3. Auswahl des Gerätes „AX-V6L“ mit dem die Synchronisierung erfolgen soll.

Mehrere Geräte können sich mit dem „AX-V6L“ synchronisieren. Jedoch kann ein Gerät sich nur mit einem anderen Gerät synchronisieren.

Der Positionierer kann also nur mit dem Roboter synchronisiert sein und nicht gleichzeitig mit weiteren Geräten.



4. Auswahl der Achsnummer des Gerätes „AX-V6L“, die mit der ersten Achse des Positionierers synchronisiert werden soll. Negative Werte bedeuten die Synchronisierung mit passiven Achsen des Gerätes. 0 wenn die Achse nicht synchronisiert sein soll.

Bewegen Sie nun die Achse 8 des Roboters, sollte sich die erste Achse des Positionierers mit bewegen.

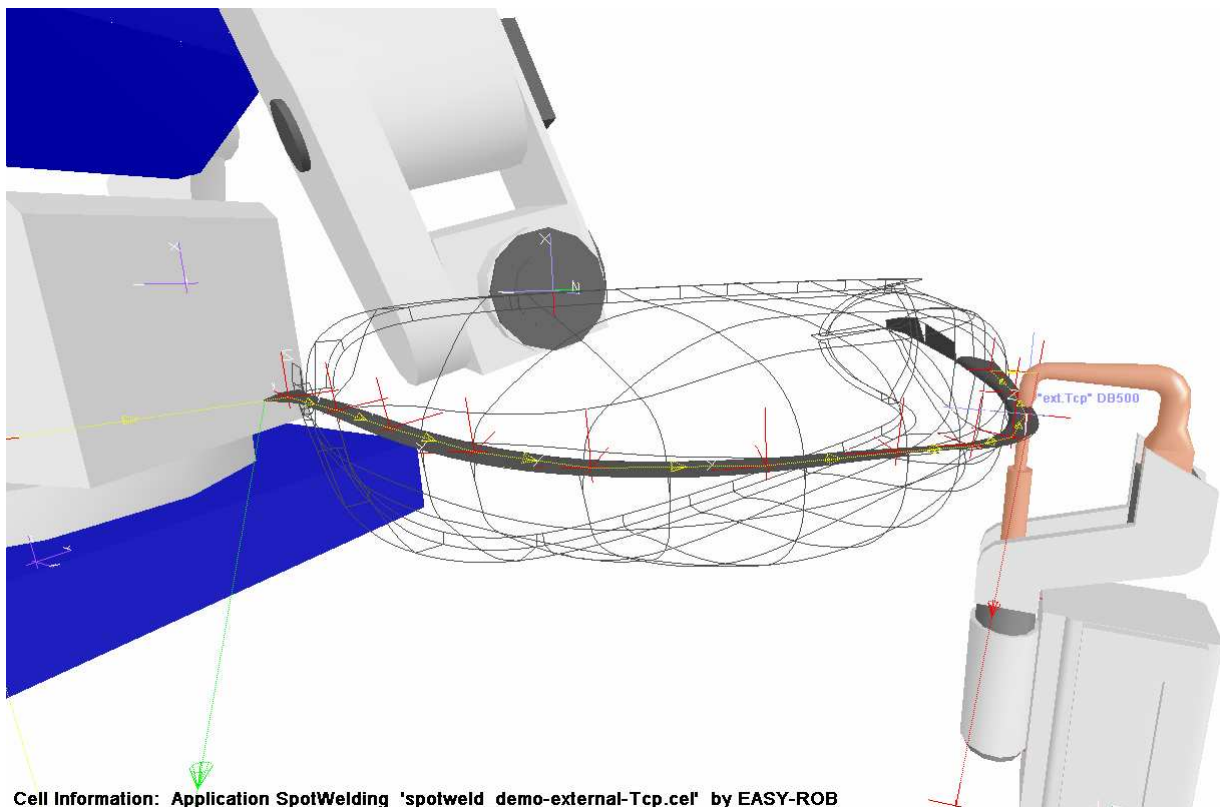
Externer TCP

Werkstückführende Bewegungen werden nun besser unterstützt, insbesondere beim Erzeugen und Ändern von Tag-Punkten am Werkstück.

Im Unterschied zur werkzeugführenden Bewegung führt der Roboter bei der werkstückführenden Bewegung das Werkstück entlang eines externen TCPs. Hierbei werden die Pfade und Tag-Punkte wie gewohnt am Werkstück definiert und anschließend die Lage des externen TCPs festgelegt. Der Modus „externer TCP“ wird mittels ERCL Kommando **„ERC EXT_TCP ON/OFF“** aktiviert bzw. deaktiviert, gefolgt von Verfahrbefehlen wie PTP, LIN, CIRC, MOVE oder Along.

Mit dem ERPL Befehl **„EXT_TCP XYZ ABC“** oder **„EXT_TCP tagname“** wird die Lage des externen TCPs festgelegt. Im Menu: „Robotics > cRobot Ext Tcp Data“ kann die Lage manuell geändert werden. Im Online Output Data Dialog wird in „current Motion Data“ mit „External TCP ON“ angezeigt, dass der Modus externer TCP aktiv ist. Ein Doppelklick auf diese Zeile aktiviert und deaktiviert den externen TCP. In der 3-D Szene wird ein aktiver externer TCP durch ein hellblaues Koordinatensystem mit Beschriftung dargestellt.

Es ist sinnvoll die Tooldaten null zu setzen, während der externe TCP aktiv ist.



Cell Information: Application SpotWelding 'spotweld_demo-external-Tcp.cel' by EASY-ROB

Beispiel: „.\ApplicationLib\SpotWelding\spotweld_demo-external-Tcp.cel“

EASY-ROB™ Lizenz Manager als Dienst

Der Lizenz Manager kann nun als Dienst installiert werden, was die Administration vereinfacht.

Bei einigen Anwendern läuft der EASY-ROB™ Lizenz Manager schon seit seiner Verwendung als Dienst. Ist der Lizenz Manager als Dienst installiert, wird er beim Starten des Rechners automatisch gestartet und beim Herunterfahren beendet. Der Lizenz Manager ist auch dann aktiv, wenn kein Benutzer angemeldet ist.

Zur Installation des Lizenz Managers als Dienst, starten Sie die Anwendung „er_LMNGR_InstSrv.exe“ aus dem Installationsverzeichnis des Lizenz Managers. (z.B. c:\Programme\EASY-ROB\EASY-ROB LicenseManager). Weitere Hinweise lesen Sie bitte in den beiliegenden Dokumenten „EASY-ROB-Lmnggr-Install-Service.pdf“ und „Applikationen als Dienste einrichten.pdf“.



Anwendung: „er_LMNGR_InstSrv.exe“

Klicken Sie „Install“ um den LizenzManager als Dienst zu installieren. Hierbei werden 2 Batchdateien („er_LMNGR_InstSrv.bat“ und „er_LMNGR_InstSrv_Remove.bat“) erzeugt, wobei die erste sofort ausgeführt wird.

Mit dem automatischen Ausführen der Batchdatei „er_LMNGR_InstSrv.bat“ wurde der Dienst mit dem Namen „**EASY-ROB-LMNGR-SRV**“ erzeugt und der Registry entsprechende Einträge hinzugefügt.

Er werden Administrator-Rechte vorausgesetzt.

Genauere Taktzeitberechnungen

AUTO_ACCEL

Unter realen Bedingungen wurden Praxistests in Form von echten Taktzeitmessungen durchgeführt. Die Ergebnisse werden nun in der EASY-ROB™ Bewegungsplanung vollständig berücksichtigt. Taktzeiten lassen sich dadurch genauer abschätzen.

Entscheidend für die resultierende Taktzeit sind die im Verfahrensprogramm vorgegebenen Geschwindigkeiten und Beschleunigungen. Bei PTP Bewegungen werden diese beispielsweise mit den Befehlen SPEED_PTP_OV %-Wert und ACCEL_PTP_OV %-Wert vorgegeben. Der %-Wert gibt den prozentualen Anteil der maximalen Achsgeschwindigkeiten und -Beschleunigungen des Roboters an, welche als Attribute vorliegen und bei Bedarf geändert werden können. Für den Anwender ist es teilweise schwierig korrekte Beschleunigungswerte vorzugeben, da Herstellerangaben oft nicht ausreichend vorliegen.

Mit dem neuen Befehl **AUTO_ACCEL ON** werden in Abhängigkeit der vorgegebenen Geschwindigkeiten die Beschleunigungen automatisch berechnet. Als Berechnungsgrundlage dienen die am realen Roboter durchgeführten Messungen. Die Berechnungen sind für PTP-, LIN- und CIRC Bewegungen gültig, bei denen im Zielpunkt angehalten wird.

Im folgenden Beispiel „.\Tutorial\Proj_example_erpl\functions_auto_accel.cel“ wurden vier Fälle simuliert.

Versuch	Speed_PTP_OV	Accel_PTP_OV	AUTO_ACCEL	Taktzeit in s
1	50	100	OFF	1.836
2	50	nicht verwendet	ON	1.769
3	100	100	OFF	1.630
4	100	nicht verwendet	ON	1.106

Tabelle 1 „Auto_Accel“

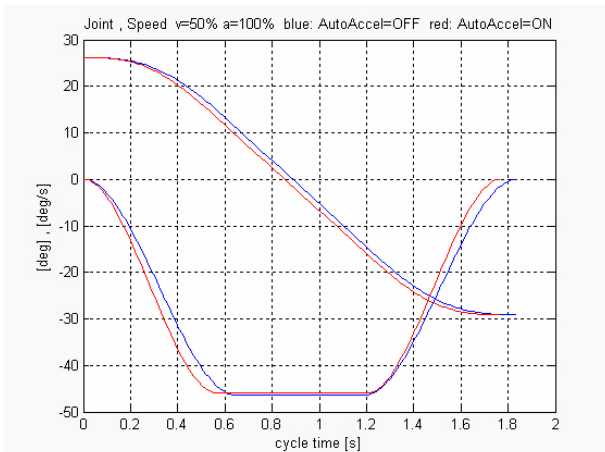
Um die Achswinkel, -geschwindigkeiten und -beschleunigungen in jedem Simulationstakt (10ms) zu speichern wurde der Befehl „ERC STATUS_OUTPUT ON 1 AUTOACCEL_50_100_off.dat -1“ verwendet. Die Ergebnisse werden so in der Text-Datei „AUTOACCEL_50_100_off.dat“ gespeichert, die mit Matlab grafisch „taktzeit_auto_accel.m“ dargestellt werden. Der Roboter fährt PTP von T_1 nach T_2.

Ergebnisse

Wie zu erwarten ist, ergibt sich bei AUTO_ACCEL OFF im Fall 3 mit Speed_PTP_OV 100% eine kürzere Taktzeit von 1.630s als im Fall 1 (1.836s) mit Speed_PTP_OV 50%. Allerdings ist die Taktzeit nur um ca. 11% geringer, was darauf hindeutet, dass die vorgegebenen Geschwindigkeiten wegen einer zu kleinen Beschleunigung oder einer zu kurzen Distanz (Abstand Start- Zielpunkt) nicht erreicht werden konnte.

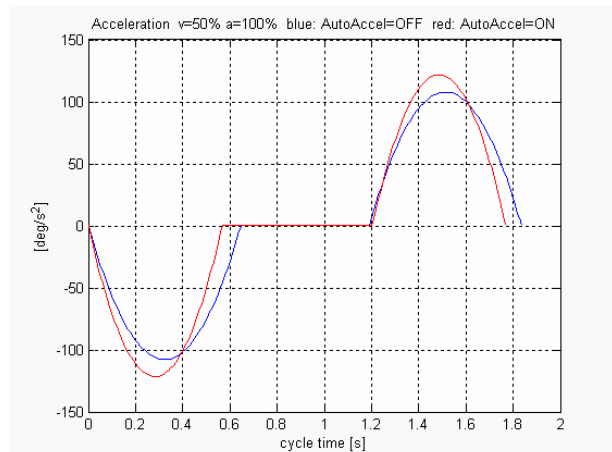
Im Fall 2 und 4 sind mit AUTO_ACCEL ON die vorgegebenen Beschleunigungen in Abhängigkeit der vorgegebenen Geschwindigkeiten automatisch berechnet worden. Vergleicht man Fall 1 mit 2 so weicht die Taktzeit um weniger als 4% ab. Wird die Geschwindigkeit jedoch auf 100% erhöht, ergibt sich im Fall 4 eine Taktzeit von nur 1.106s, die sehr stark von der im Fall 3 abweicht (32%). Da sich die Abstände der Zielpunkte nicht geändert haben, ist offensichtlich eine höhere Beschleunigung für die kürzere Taktzeit maßgebend.

Vergleich: Auto_Accel OFF/ON bei v=50% und 100%



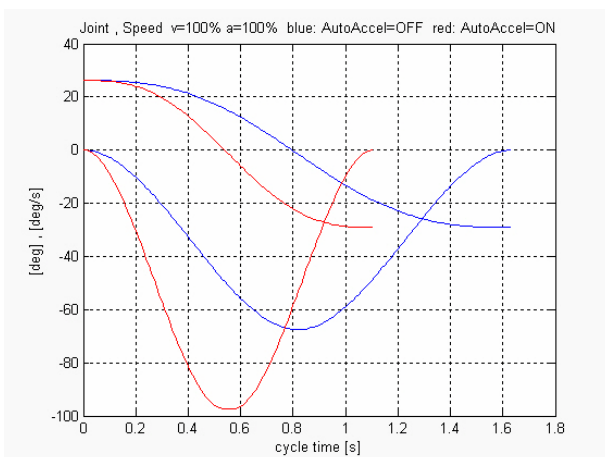
Achswinkel- und Geschwindigkeitsverlauf Achse 1 „AutoAccel_qv_50_100.jpg“

blau: v=50% a=100% Auto_Accel OFF 1.836s
rot: v=50% a=- Auto_Accel ON 1.769s



Beschleunigungsverlauf Achse 1 „AutoAccel_a_50_100.jpg“

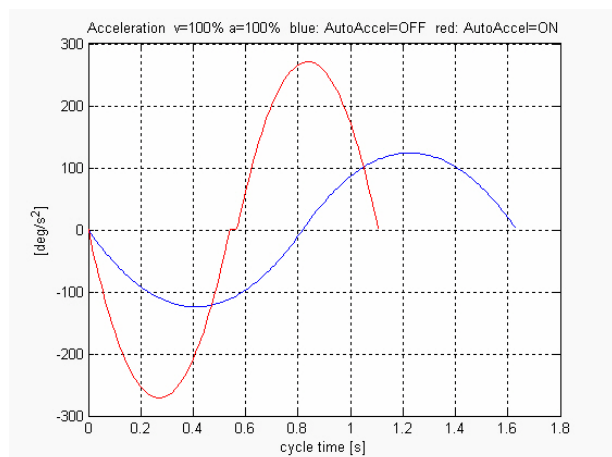
Die berechneten Beschleunigungen stimmen mit den vorgegebenen Beschleunigungen annähernd überein.



Achswinkel- und Geschwindigkeitsverlauf Achse 1 „AutoAccel_qv_100_100.jpg“

blau: v=100% a=100% Auto_Accel OFF 1.630s
rot: v=100% a=- Auto_Accel ON 1.160s

blau: v_max = 67.6%
rot: v_max = 97.5%



Beschleunigungsverlauf Achse 1 „AutoAccel_a_100_100.jpg“

blau: a_max = 124.0%
rot: a_max = 271.0%

Die berechneten Beschleunigungen sind erheblich größer als die vorgegebenen Beschleunigungen, wodurch sich eine erheblich kürzere Taktzeit ergibt.

ACCSET

Mit dem neuen Befehl ACCSET kann die vorgegebene Beschleunigung zusätzlich beeinflusst werden. Der Befehl wird beim Handhaben von zerbrechlichen oder schweren Lasten verwendet. Damit ist eine verzögerte Beschleunigung erreichbar.

ACCSET hat die 2 Argumente

Acc - Beschleunigung und Verzögerung als Prozentwert der normalen Werte

Ramp - Anstiegswert von Beschleunigung und Verzögerung als Prozentwert der normalen Werte

Im folgenden Beispiel ".\Tutorial\Proj_example_erpl\functions_accset.cel" wurden die Argumente in den Stufen 20%, 50%, 80% und 100% variiert.

ACCSET wirkt unabhängig vom Befehl AUTO_ACCEL. Somit werden auch Beschleunigungen die mit Accel_PTP_OV vorgegeben werden entsprechend verzögert.

Im folgenden Beispiel ".\Tutorial\Proj_example_erpl\functions_accset.cel" wurden vier Fälle simuliert.

Versuch	AUTO_ACCEL	ACCSET	Taktzeit in s
1	ON	20% 20%	1.815
2	ON	50% 50%	1.380
3	ON	80% 80%	1.180
4	ON	100% 100%	1.106

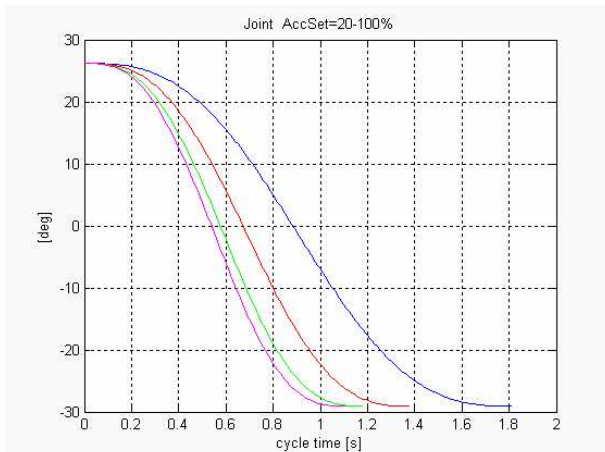
Tabelle 2 „AccSet“

Um die Achswinkel, -geschwindigkeiten und -beschleunigungen in jedem Simulationstakt (10ms) zu speichern wurde der Befehl „ERC STATUS_OUTPUT ON 1 ACCSET_20_20_on.dat -1“ verwendet. Die Ergebnisse werden so in der Text-Datei „ACCSET_20_20_on.dat“ gespeichert, die mit Matlab grafisch „taktzeit_accset.m“ dargestellt werden. Der Roboter fährt PTP von T_1 nach T_2.

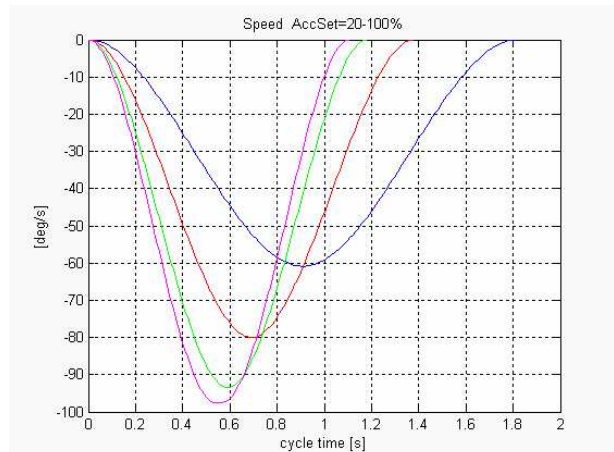
Ergebnisse

Wie zu erwarten war, vergrößert sich die Taktzeit mit kleineren Werten für die Argumente Acc und Ramp

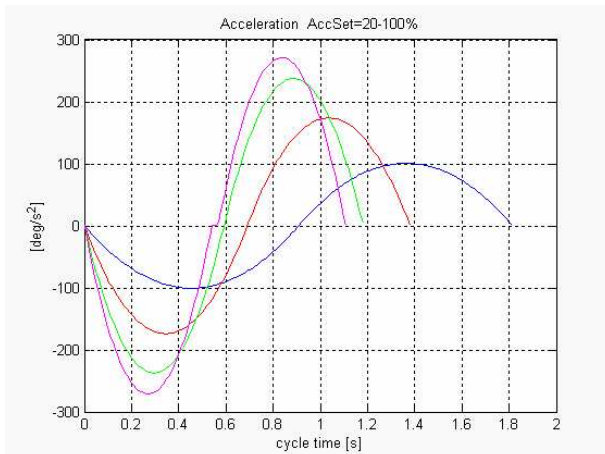
Vergleich: ACCSET bei 20%, 50%, 80% und 100%



Achswinkelverlauf Achse 1 „AccSet_q_20_50_80_100.jpg“



Geschwindigkeitsverlauf Achse 1
„AccSet_v_20_50_80_100.jpg“



Beschleunigungsverlauf Achse 1
„AccSet_a_20_50_80_100.jpg“

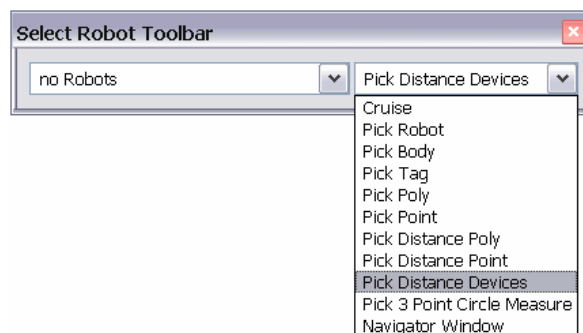
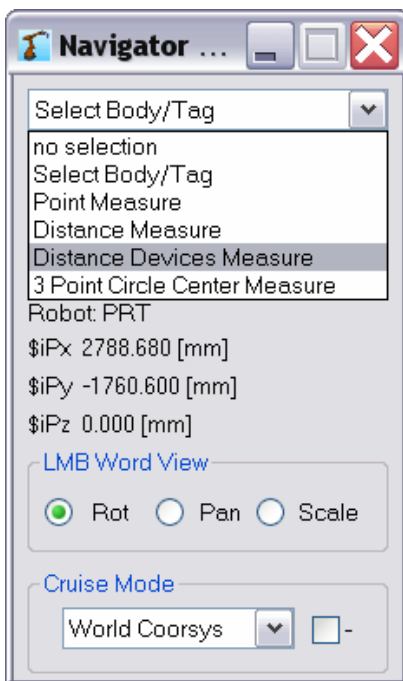
Die maximalen Geschwindigkeiten und Beschleunigungen verringern sich bei kleineren Acc- und Ramp-Werten, wodurch die Taktzeit steigt.

blau:	Acc=20%	Ramp=20%	1.815s
rot:	Acc=50%	Ramp=50%	1.380s
grün:	Acc=80%	Ramp=80%	1.180s
violett:	Acc=100%	Ramp=100%	1.106s

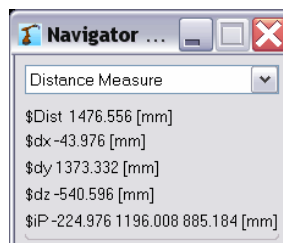
Messen

Abstände zwischen Devices messen

Um den Abstand zwischen zwei Geräten zu messen, öffnen Sie das Navigator Window und wählen „Distance Devices Measure“ aus und „picken“ anschließend die gewünschten Geräte mit der linken Maustaste an. Alternativ kann die Funktion im Robot Toolbar ausgewählt werden.

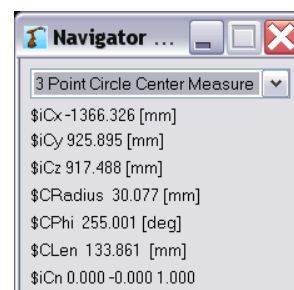
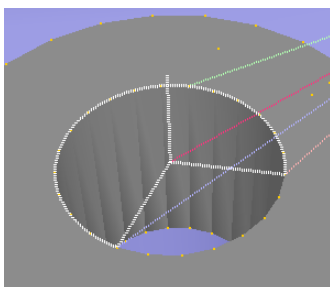


Die Abstandswerte werden bzgl. Weltkoordinaten (X, Y, Z, Abstand) im Navigator Window angezeigt.



Kreismittelpunkt messen

Um den Kreismittelpunkt aus drei Punkten zu messen, wählen Sie im Navigator Window „3 Point Circle Measure“ aus und „picken“ anschließend drei Punkte mit der linken Maustaste an.



Kreismittelpunktkoordinaten
 $\$iCx$, $\$iCy$, $\$iCz$
 Radius
 $\$CRadius = 30.077$ mm
 Winkel
 $\$CPhi = 255^\circ$
 Segmentlänge
 $\$CLen = 133,86$ mm
 Normalenvektor
 $\$iCn = 0 \ 0 \ 1$

Erzeugen Sie mit der mittleren Maus-Taste einen neuen Tag-Punkt. Platzieren Sie diesen neuen Tag-Punkt an den gemessenen Kreismittelpunkt indem Sie die Strg-Taste kurz drücken.

ERPL-Befehle

JUMP_TO tagname

JUMP_TO X Y Z A B C [m,deg]

Springt zur Zielposition (tagname - Name des Ziel-Tagpunktes, XYZ - Position, ABC - Orientierung)

JUMP_TO_AX q1 .. qn [m,deg]

Springt zur Ziel-Achswinkelstellung (q1..qn - Ziel Joint/Axis)

EXT_TCP tagname

EXT_TCP X Y Z A B C [m,deg]

Lage des externen TCPs

tagname - Name des Tagpunktes

XYZ - Position, ABC - Orientierung bzgl. Weltkoordinatensystem

AUTO_ACCEL ON [OFF, POS, ORI, AX]

Automatische Berechnung der Beschleunigung in Abhängigkeit der programmierten Geschwindigkeit.

AX – Berechnung für PTP Bewegungen

POS – Berechnung für CP Bewegungen für Position

ORI – Berechnung für CP Bewegungen für Orientierung

ON – Berechnung für PTP und POS und ORI

OFF – Berechnung deaktiviert

ACCSET Acc Ramp

Verzögerung der Beschleunigung.

Die Argument Acc und Ramp werden prozentual im Bereich von 20% - 100% angegeben.

Acc – Beschleunigung und Verzögerung als Prozentwert der normalen Werte

Ramp – Anstiegswert von Beschleunigung und Verzögerung als Prozentwert der normalen Werte

ERCL-Befehle

ERC EXT_TCP ON/OFF

Ein- und Ausschalten der werkstückführenden Bewegung, externer TCP

ERC CELL_INFO_SHOW ON/OFF

Ein- und Ausschalten der Zellen-Informationszeile

ERC CELL_INFO text

Text für Zellen-Informationszeile

Parser-Funktionen

sim_time()

Returnwert: aktuelle Simulationszeit [s]

API-Funktionen

Neuer Header „er_capi_types.h“

In der neuen Header-Datei „./er_dvlp/er_capi_types.h“ sind sämtliche Konstanten- und Typendefinitionen deklariert. Der Header wird von „./er_dvlp/er_dvlp.h“ inkludiert. Die Header-Dateien „./er_dvlp/er_dvlp.h“ und „./er_dvlp/er_dvlp_ext.h“ beinhalten somit nur noch die Funktionsdeklarationen.

USER_IO_DIALOG

- `int _info_line_msg_i (int moni, char *s, int *vi, int n, int scal)`

```
// Integer-Vektor-Ausgabe im Message Window
moni      0- message window, 1- message window und moni_msg.txt
*s        Info-Text
*vi       Vector
n         Dimension von vi
scal      Skalierung
```

- `int _info_line_msg_T_vec (int moni, char *s, frame *T)`

```
// Frame-Ausgabe aus Beschreibungs-Vektor Pxyz, Rxyz im Message Window
moni      0- message window, 1- message window und moni_msg.txt
*s        Info-Text
*T        Frame
```

SIM_ERPL – Simulation ERPL

- `int *inq_single_cmd_ext_Tcp_idx (void)`

```
// Status Externer TCP aktiviert=1 oder deaktiviert=0 bei Einzelsatzausführung
```

- `int *inq_single_cmd_c_device_idx (void)`

```
// Aktueller Device Index bei Einzelsatzausführung
```

DEVICES

- **int *inq_device_sync_ref_sys_type (void)**

// Referenztyp für Synchronisation
 // REF_NO_REF = 0 aktuelles Device nicht synchronisiert
 // REF_JNT = 9 aktuelles Device mit Achse synchronisiert

Return Pointer to value

- **char *inq_device_sync_ref_sys_type_name (void)**

// Name des Referenztyps für Synchronisation
 // „Ref_no_Ref“ aktuelles Device nicht synchronisiert
 // „Ref_Jnt“ aktuelles Device mit Achse synchronisiert

Return String

- **long *inq_device_sync_ref_sys_grp_uid (void)**

// Unique ID des Referenz Devices, sonst NULL

Return Pointer to value

- **int *inq_device_sync_ref_sys_jnt_sync_idx (void)**

// Index der Achsen des Referenz Devices mit denen das aktuelle Device synchronisiert ist.
 // Index = 0 Achse ist nicht synchronisiert
 // Index < 0 Achse ist mit passiver Achse des Referenz Devices synchronisiert
 // Index > 0 Achse ist mit aktiver Achse des Referenz Devices synchronisiert

Return Index-Vector

- **int Device_Sync_by_name (int new_reference_type, char *new_reference_device_name=NULL, int *new_reference_jnt_sync_idx=NULL)**

// Bestimmung der Synchronisations-Beziehungen mittels Referenz-Device-Namen
 new_reference_type REF_NO_REF = 0, REF_JNT = 9
 new_reference_device_name Name des Referenz Devices
 new_reference_jnt_sync_idx Sync Index Vector der Achsen des Referenz Devices (REF_JNT)

Return 0 – OK, 1 – Error

- `int Device_Sync_by_idx (int new_reference_type, int new_reference_device_idx=0, int *new_reference_jnt_sync_idx=NULL)`

// Bestimmung der Synchronisations-Beziehungen mittels Referenz-Device-Index

<code>new_reference_type</code>	REF_NO_REF = 0, REF_JNT = 9
<code>new_reference_device_idx</code>	Index des Referenz Devices
<code>new_reference_jnt_sync_idx</code>	Sync Index Vector der Achsen des Referenz Devices (REF_JNT)

Return 0 – OK, 1 – Error

- `int Device_Sync_by_uid (int new_reference_type, long new_reference_device_uid=0, int *new_reference_jnt_sync_idx=NULL)`

// Bestimmung der Synchronisations-Beziehungen mittels Referenz-Device-Unique ID

<code>new_reference_type</code>	REF_NO_REF = 0, REF_JNT = 9
<code>new_reference_device_uid</code>	Unique ID des Referenz Devices
<code>new_reference_jnt_sync_idx</code>	Sync Index Vector der Achsen des Referenz Devices (REF_JNT)

Return 0 – OK, 1 – Error

ROB_KIN – Robot Kinematics

- `char *inq_kin_chain_type_activ (void)`

// 'C'	aktive Achse ist in der kinematischen Kette (standard)
// '-'	aktive Achse ist separiert und nicht in der kinematischen Kette (neu)

- `frame *inq_achs_T0_activ (void)`

// Transformation zur aktiven Achse für den Fall dass die aktive Achse
// separiert und nicht in der kinematischen Kette ist '-'

- `char *inq_robot_fln_name (void)`

// Dateiname des aktuellen Roboters, sonst NULL

- `char *inq_kin_chain_type_passiv (void)`

// ersetzt `char *inq_kin_chain_type (void)` // obsolete

// 'C'	passive Achse ist in der kinematischen Kette
// '_'	passive Achse ist nicht in der kinematischen Kette
// '-'	passive Achse ist separiert und nicht in der kinematischen Kette

- `char *inq_kin_calc_passiv (int passiv_jnt_no)`
 // ersetzt `char *inq_kin_calc (int passiv_jnt_no) // obsolete`
 passiv_jnt_no Index passive Achse, nullbasiert
 Return Formel-String für mathematische Abhängigkeit

- `int *inq_kin_attach_dof_passiv (void)`
 // ersetzt `int *inq_kin_attach_dof (void) // obsolete`
 // z.B. „pJ3 RYC2-“ pass. Achse 3, Roty, in der kin. Kette, „attached“ an aktive Achse 2
 // Hat die Kinematik beispielsweise 3 passive Achsen RY_1-, RY_3-, RYC2- ergibt
 // sich der Index-Vector zu [1, 3, 2]

 Return Index-Vector, Index der aktiven Achse an der die passive Achse ‚attached‘ ist.

MOP_PATH – Motion Planner Path

- `int *inq_ipo_path_no_decel (void)`
 // 0 – Endgeschwindigkeit = 0,
 // 1 – Kein Bremsen Endgeschwindigkeit = Bahngeschwindigkeit (zone>0 setzen)

 Return Pointer to value

- `float *inq_ipo_path_zone (void)`
 // Überschleifparameter,
 // „Zone = 0“ ermöglicht das exakte Anfahren einer Zielstellung
 // „Zone > 0“ Interpolationstakt übergeht die Zielstellung

 Return Pointer to value

- `float *inq_ipo_path_AccSet (void)`
 // Verzögerung der Beschleunigung.
 // Die Argumente Acc und Ramp werden prozentual im Bereich von 20% - 100% angegeben
 // Acc Beschleunigung und Verzögerung als Prozentwert der normalen Werte
 // Ramp Anstiegswert von Beschleunigung und Verzögerung als Prozentwert der normalen Werte

 Return Pointer to values [Acc , Ramp]

- `int *inq_ipo_path_AutoAccel (void)`
 - // Automatische Berechnung der Beschleunigung in Abhängigkeit der
 - // programmierten Geschwindigkeit
 - // ON = 7 – Berechnung für PTP, POS und ORI
 - // POS = 1 – Berechnung für CP Bewegungen für Position
 - // ORI = 2 – Berechnung für CP Bewegungen für Orientierung
 - // AX = 4 – Berechnung für PTP Bewegungen
 - // OFF = 0 – Berechnung deaktiviert

Return Pointer to value

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

Vertriebsbüro - Nord

Gregor Hölting

West I 17, 48324 Sendenhorst, Germany

Tel. +49 (0) 2506 81 65 73

Fax. +49 (0) 2506 81 65 74

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