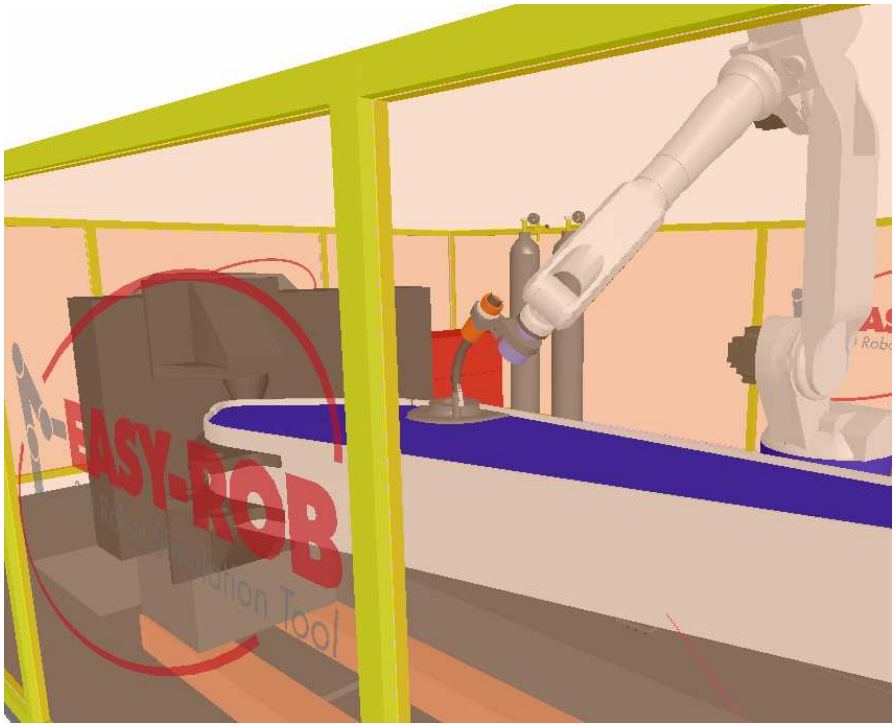
The new Version

EASY-ROB™ V6.0



JULY 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™

Table of contents

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™ completely available as 64-Bit Version**
  The complete EASY-ROB™ Product Suite in Version 6.0 (**Multi-Program**- and **Single-Robot** Version, **DLL** Version, **Robotics Simulation Kernel** and **Viewer** Version) is now available without any limitations as 64-Bit Version for Window® 7 64-Bit.

- **Enhanced support with TeamViewer**
  In order to support our customers much better, we provide you with the EASY-ROB™ TeamViewer Quick Support**.** A lot of problems can be solved much faster, by joining and watching either your or our EASY-ROB™ session.

- **New robot models**
  New robot models from ABB, Comau, Kuka, Motoman, Stäubli and Universal Robots (UR-5 und UR-10) have been added to the robotic library. Actually more than 450 robots are available.

- **Robot post processors**
  API programming examples are now prepared for the robot languages from ABB, Kuka, OTC, Comau, b+m and Fanuc. Appropriate adjustments allow the user to make quick changes and personal extensions.

- **Full-Synchro-PTP and SLEW**
  The motion-type Synchro-PTP has been upgraded to Full-Synchro-PTP. That means, the TCP Trace is now completely independent from the programmed speeds, accelerations and override.
  Additionally the new motion type „SLEW“ for asynchronous PTP has been added. The axes will not be time-synchronized in this motion type.

- **Optimizations**
  Identical geometries will be cloned and loaded only once, which saves a lot of resources and reduces the loading time of work cells and devices.
  Every DLL used in EASY-ROB™ will be linked dynamically. The names of the user-specific DLLs can be defined in the environment file „easy-rob.env“.

- **EASY-ROB™ Robotics Simulation Kernel**
  New examples of the use of EASY-ROB™ Robotics Simulation Kernel are given. This should help you with the integration into your own applications.
  In addition to the licensing-process with the WibuKey-Dongle and Hardware-Number, the kernel can be licensed using the EASY-ROB™ License Manager.

- **New Robot Jog Mode „TCP Base“**
  The robot TCP can be jogged now with reference to his robot base. This corresponds - including the „TCP Tool“, „TCP World“ and „Robot Joints“ modes – to every manual moving type found on the programming device of different robot controls.

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

- **Manipulation of paths and tags**
  New possibilities to manipulate paths and tags have been added. For example: Mirroring of paths, alignment of tag points to an axis, partial copy of tag-positions, reversion of path-directions and redefining of work-objects.
  Additionally the length, angle and distance to previous and following tags can be displayed.
  External axe-values (of e.g. a positioner or a rotary tilting table) can be calculated automatically in dependence of the path-length by using „AutoCalc“.

- **New kinematics type: JET Robots (gantry robots)**
  JET robots with the ID 127 (respectively 128 for A2A3 coupling) using the serial structure TyRyy:Rxyx or TyRyy:Rzyz with 4 configurations are supported and can be created manually by customers (e.g.. KR 30 JET, KR 60 JET).
  Two- and three-axis gantry kinematics (ID 133) can be created in the following combinations: Txyz, Tyxz, Tzxy, Tzyx, Txzy, Tyzx bzw. Txz, Tyz, Tzx, Tzy, Txy, Tyx

- **New robot attributes**
  Active or passive joints of robots, kinematics or devices can be named now, e.g. „vertical axis“ or „transverse drive“ instead of „Joint_1“. Besides names can be assigned to configurations, like e.g. for “Kuka S'B010” or “Fanuc NUT” instead of “Config_1”.
  In addition to the turn-interval, turn-offsets can be defined for every joint. This is necessary for the proper synchronization with the robot-controller.

- **Exporting paths into native robot programs**
  The generic export of paths into native robot programs has been enhanced for the control types of ABB, KUKA, COMAU, Fanuc, b+m and OTC. More control types can be integrated on our customers wish.

- **Collision**
  The new collision-algorithm „PQP” allows you to define tolerances. That means that collision will be indicated if two bodies approach each other and reach a minimal distance. This tolerance can be set individually for each body.

- **New ERPL and ERCL commands**
  SLEW, SLEW_REL, SLEW_AX, SLEW_AX_REL for asynchronous PTP
  “ERC GRAB_TO” to re-attach a device to other devices
  “ERC COLLISION DISTANCE …“ to define the collision-tolerance of single geometries

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, Method-Class ER_CAPI**
  Many new API functions for individual product customization and special solutions has been added. These new functions allow to control EASY-ROB™ out of an own application or to exchange data in a bidirectional way.
  The method-class ER_CPAI serves as interface for the EASY-ROB™ **Multi-Program** and EASY-ROB™ **DLL** version as well as for the extensions **API-INV**, **API-IPO**, **API-DYN**, **API-UserDLL**, **API-PostProc** and **API-Sensors**.

- **Other Improvements**
  - Teach-Window with new dialogue and all motion commands
  - Scalable Device Manager dialogue
  - More resolutions for AVI-Recorder available
  - The sensitivity and threshold value of a 3D Space Mouse can be set in the environment file
  - New parser-functions allow access to kinematic robot-lengths a more
  - Advanced Status Output for the output of e.g. axis values in every simulation step

The new version is available free of charge for all customers with a valid license key for EASY-ROB™ V6.0. For customers using older versions, it will be possible to purchase an update. We would like to thank you for your suggestions and ideas in advance.

Thank you

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

Finally, after 12 years the new EASY-ROB version 6.0 with many new features and improvements has been released.

Certainly many customers and prospective desire a simplification of our software. Everyone wants to get a quick and reliable idea of feasibility and cycle time using a simulation-based tool for robot work cell-planning. At the same time, however, the complexity and requirements of the system are raising. So we are still facing a dilemma: “should we develop a easy to handle tool with less functionality or should it be a very complex software with many possibilities?” We have chosen to take the hard way and decided to provide our customers and OEM-partners, using the EASY-ROB™ Product Suite with a lot of functionality. This includes the consistent development of the programming interfaces (ER_CAPI), the EASY-ROB™ DLL Version as well as the EASY-ROB™ Robotics Simulation Kernel.

In our opinion a minimum of robotics know-how is required for the effective operation of EASY-ROB™. This makes sure a user is able to understand which robot-settings have an effect on trace, orientation or cycle-time. Also important is a three-dimensional imagination as well as mathematical skills to understand the importance of 3D-coordinates (e.g. Euler angles) and transformations in the space. Users of the API, of course require programming skills in C/C++.

To solve this problem, we offer a one-day robotics-schooling in addition to the regular EASY-ROB™ training.

Nevertheless it is our goal to simplify the handling of EASY-ROB™ as best as possible.

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 with TeamViewer

In order to support our customers much better, we provide you with the EASY-ROB™ TeamViewer Quick Support**.** A lot of problems can be solved much faster, by joining and watching either your or our EASY-ROB™ session.

Support with TeamViewer Quick Support

Download TeamViewer QS

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

The program „teamviewerqs_er_de.exe“ can be started without installation and administrative rights. It allows us to help you spontaneously no matter if you have problems or a question to a product.

Presentation with TeamViewer Quick Join

Download TeamViewer QJ

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

The program „teamviewerqj_er_de.exe“ can be started without installation and administrative rights. It allows us to present you with our products.

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

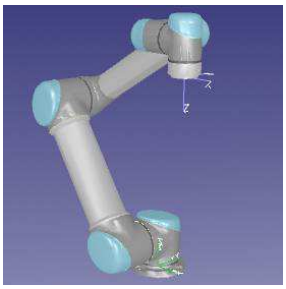
New robot models

The robotic library has grown again. New models from ABB, Comau, Kuka, Motoman, Stäubli and Universal Robots (UR-5 and UR-10) have been added.

Actually there are more than **450 Robots** available.

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

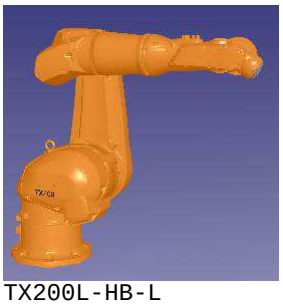

NJ-130-2,6


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

Stäubli Roboter

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



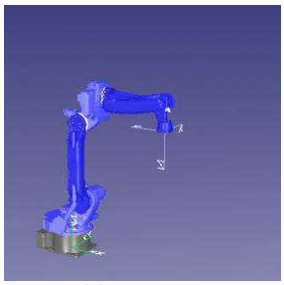
TX200L-HB-L

TP80

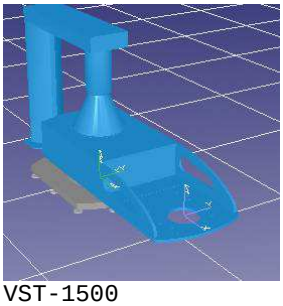
Motoman

MA1900
MA1800
MA1400
VST-600
VST-1500

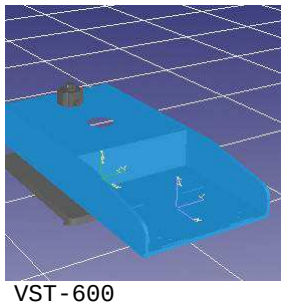
MA1900



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

Robot post processors (API)

API programming examples are now prepared for the robot languages from ABB, Kuka, OTC, Comau, b+m and Fanuc. Appropriate adjustments allow the user to make quick changes and personal extensions.

The option API-Post-Proc allows to develop own post processors for EASY-ROB™. Post processors for ABB, Kuka, b+m, Fanuc and OTC are prepared in the Visual Studio 2008 example project “er_post.sln”. The created DLL “er_post.dll” will be linked dynamically.

In ERPL the post processor will be started with the command “ERC POST_PROCESS LANGUAGE_KEY flnname“

You can choose e.g. “KUKA”, “OTC” or “Fanuc” for the LANGUAGE_KEY. These keys can be extended at will.

In the programming example only the class “ER_CPost_Base” is derived. Therefore only the major (abstract) methods have to be overwritten at the beginning.

```
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

Robot post processors (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
```

The post processor API can also write out individual languages, depending on the application and control.

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

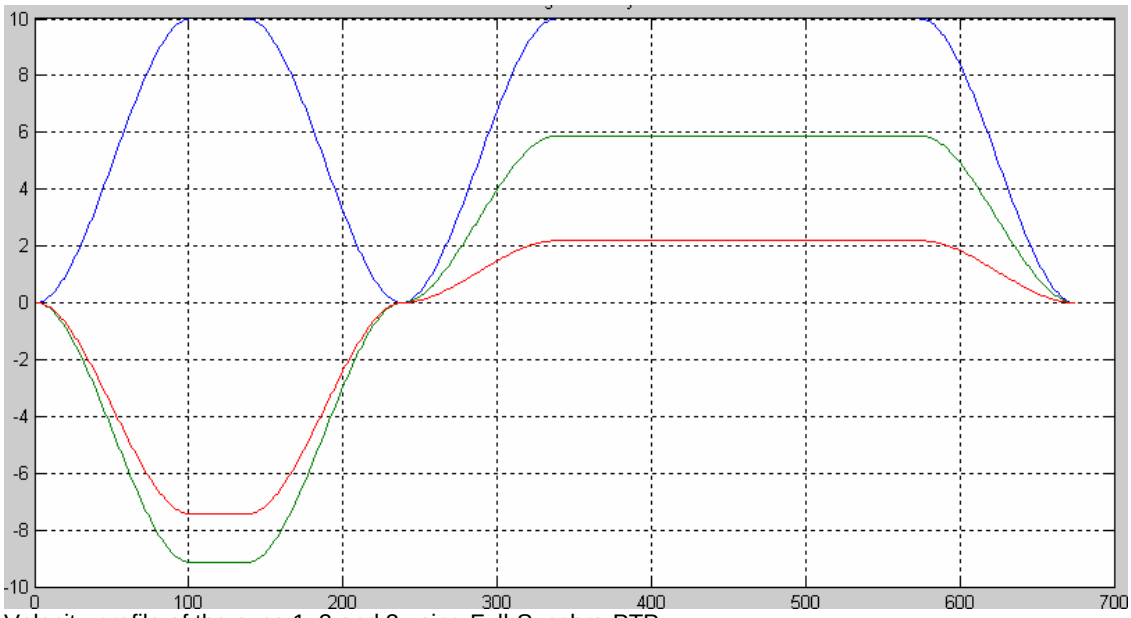
Full-Synchro-PTP and SLEW Motion

The motion-type Synchro-PTP has been upgraded to Full-Synchro-PTP. The TCP trace is now completely independent from the programmed speeds, accelerations and the override.
Additionally the new motion type „SLEW" for asynchronous PTP has been added. The Axes will not be time-synchronized in this motion type.

Full-Synchro-PTP

In the synchronous PTP all axes start and also end their movement at the same time. The so called master-axis or dominant axis is indicating the whole movement all over the time. Usually the axis with the longest movement-time is dedicated to be the dominant axis. The other axes are adjusted in time.

The Full-Synchro-PTP (also phase-synchronous PTP) assures the adjustment of the acceleration- and deceleration-phases of all involved axes to the phases of the dominant axis.



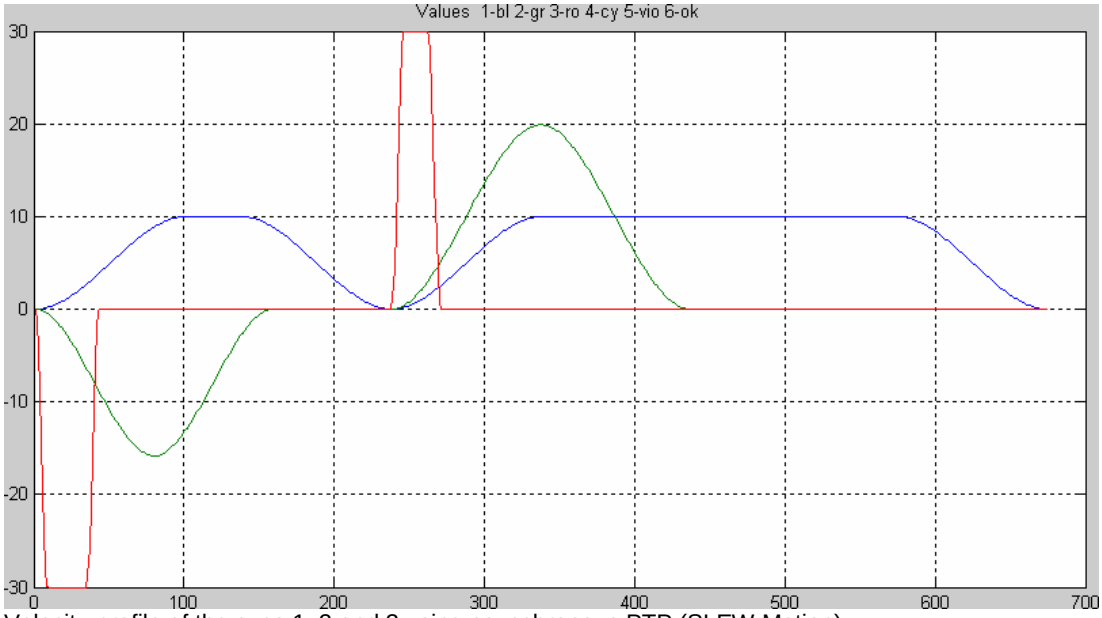
Velocity-profile of the axes 1, 2 and 3 using Full-Synchro-PTP
Axis 1 (blue) is the dominant axis and the only one, which reaches the programmed axis-speed of v = 10°/s at a = 10°/s²
The axis 2 (green) and 3 (red) will be slowed down.
Work cell: „PTP-SLEW.cel" with status output file „ER431_2_PTP.dat"

The advantage with Full-Synchro-PTP is that the resulting TCP trace of the robot is independent of the programmed axis speeds and accelerations.

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

Asynchronous PTP (SLEW)

In the asynchronous PTP (SLEW Motion) all axes start moving at the same time.
Depending on the programmed speeds, accelerations and distances the axes will reach their target positions at different times. So there is no dominant axis respectively any synchronization with the other axes.
This new Motion type “SLEW” is more suitable for “other” devices than for robots.



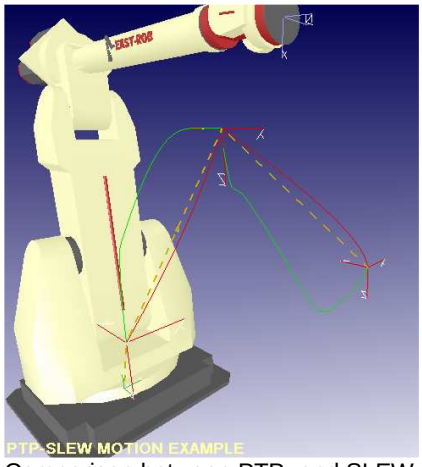
Velocity-profile of the axes 1, 2 and 3 using asynchronous PTP (SLEW Motion)
Every axis reaches its programmed axis speeds with 10°/s, 20°/s und 30°/s
The axis 2 (green) and 3 (red) reach their target position earlier. The axis 1 (blue) is the slowest axis.
Work cell: „PTP-SLEW.cel“ with status output file „ER431_2_SLEW.dat“

Example Work cell: „PTP-SLEW.cel“

The program „PTP-SLEW-ER431-2.prg“ creates two status output files „ER431_2_PTP.dat“ and „ER431_2_SLEW.dat“, which can be shown graphically using the MATLAB® files “ptp_slew.m“ and “show_ax.m“. Call ptp_slew(3), 3 = Number of displayed axes, here 1,2 and 3.

TCP-Trace:

PTP → red
SLEW → green



Comparison between PTP- and SLEW Motion “

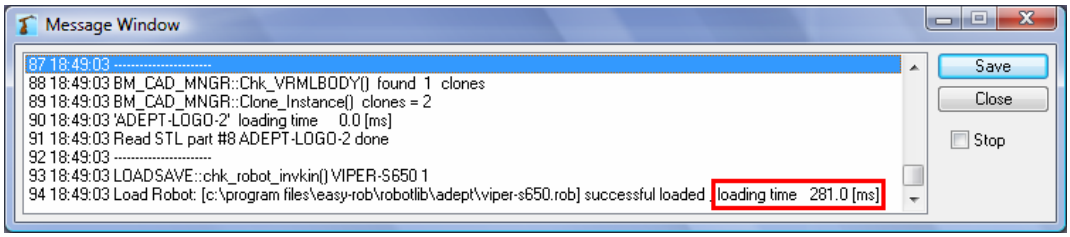
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Optimization

Reducing of loading time by cloning identical geometries

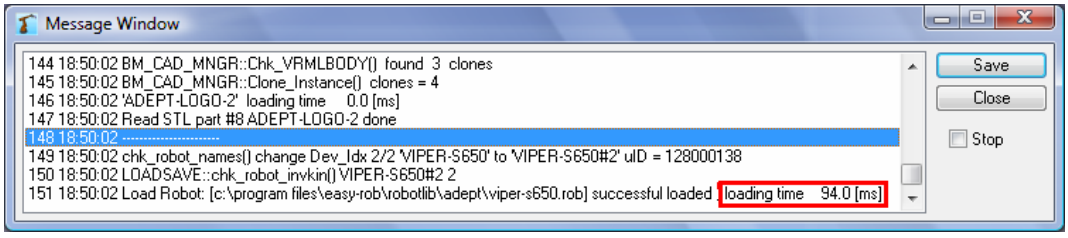
Identical geometries will be cloned and loaded only once, which saves a lot of resources and reduces the loading time of work cells and devices.
Same geometries will be cloned instead of being loaded twice. This results in an increment of performance, a shorter loading time of robots which use the same geometries and also lower memory utilization especially for complex geometries.

1. Open the Message-Windows (Ctrl + M) and load a robot



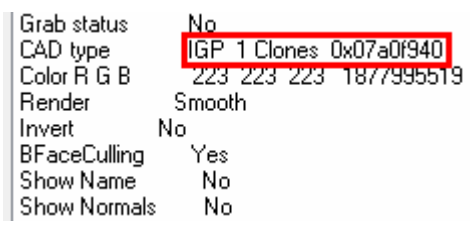
The first loading process of the robot VIPER-S650 takes a time of about 281 ms.
(Please note: the indicated loading time is a value which varies from system to system. This is only an example which shows the loading time-saving)

2. Load a second (identical) robot



If a second identical robot is loaded, the loading time takes about 94ms. The charging time of already existing geometries has been eliminated.

Hint: When identical geometries are found, the names and numbers of the created clones will be displayed in the 3D-CAD Window.



The loaded geometries of the robot can be selected in the 3D-CAD Window, where you can also see the present number of clones in the line “CAD-type”.

Hint: Attributes like name, render type, color, offset-position, collision tolerance etc. will be doubled, because each geometry still needs to have its own attributes.

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

Dynamic linking of DLLs

All DLLs which are used by EASY-ROB™ can be linked dynamically. The names of the user-specific DLLs can be set in the environment file “easy-rob.env”.

This allows starting EASY-ROB™ with an individual solution.

Environment file: “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

The EASY-ROB™ Simulation Kernel is a developers version for integration into own custom applications. The kernel does all the calculations, such as forward and inverse coordinate-transformation, motion-planning and –execution (PTP, LIN, CIRC, SLEW) for all available types of robots. To control it, only C/C++ API-functions/services for robot functionality are given.
The custom application does its own 3D-visualization, as well as the management of all geometries and handling of every loaded kinematics. The EASY-ROB™ Simulation Kernel returns a handle for every loaded kinematics.

New examples of the use of EASY-ROB™ Robotics Simulation Kernel are given. This should help you with the integration into your own applications.

In addition to the licensing-process with the WibuKey-Dongle and Hardware-Number, the kernel can be licensed using the EASY-ROB™ License Manager.

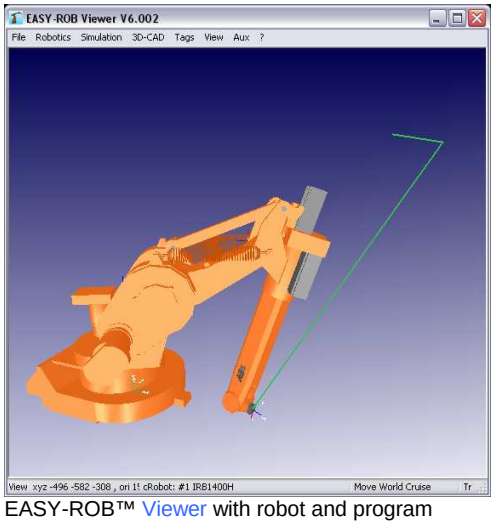
The following files belong to the EASY-ROB™ Kernel files:

- version.txt // version
- EasySimKernel.dll // Windows Dll
- EasySimKernel.lib // Library for Linker
- EasySimKernel.def // def file if needed
- er_Kernel_main.h // Header files, declaration of types, prototypes, etc.
- ipo_extax.h // Header files
- er_wibukey.dll // for WibuKey USB Dongle licensing
- EasySimKernelx64.dll // Windows Dll, 64-Bit version
- EasySimKernelx64.lib // Library for Linker, 64-Bit version
- er_wibukeyx64.dll // for WibuKey USB Dongle licensing, 64-Bit version

Kinematics example

In this simple kinematics example the TCP-position will be changed step by step until the inverse kinematics solution returns an error.

In this case the traveling ranges of axis 2 have been exceeded. Another mistake would be for example the unreachability of the target position.



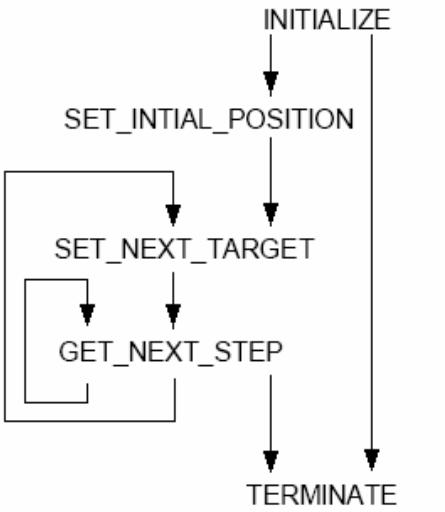
EASY-ROB™ Viewer with robot and program

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

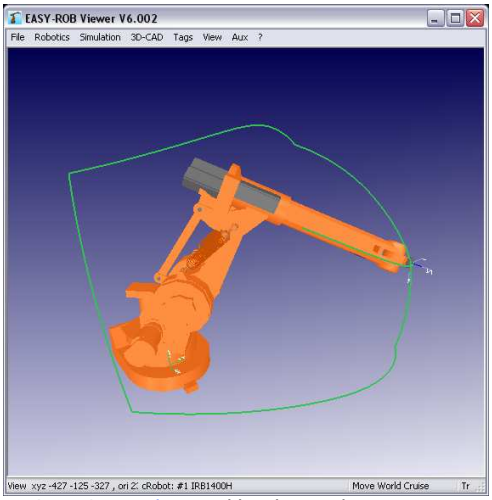
Motion Planner example for PTP, LIN, CIRC

In this Motion Planner example some target positions will be approached using the PTP, LIN or CIRC motion type.

The example shows the determination of speeds and interception of errors.



Principal RRS-Services (Source: Fhg-IPK-Berlin)



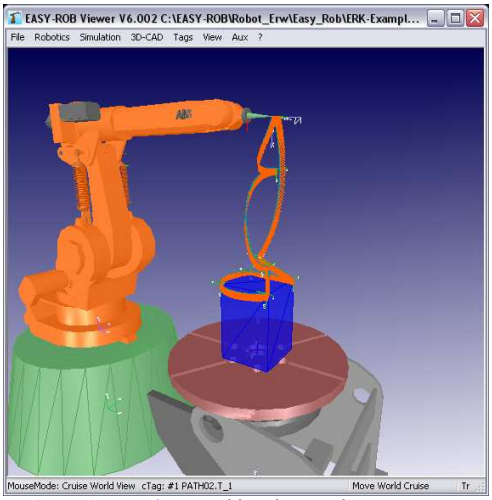
EASY-ROB™ Viewer with robot and program

Motion Planner example with synchronized positioner

In this example the robot motion will be synchronized with the external positioner.

The target positions are defined with respect to the turn-table or the work object.

The example shows how devices - placed in space - are attached together and synchronized. Additionally it shows how external axis-values can be set for specific targets.



EASY-ROB™ Viewer with robot and program

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

Modules and options

The EASY-ROB™ Robotics Simulation Kernel is divided into modules, which makes it possible to book successive options.

The basic module consists of „erk01 - ERK Kinematics - Single Robot" and includes forward and inverse coordinate transformation.

The basic module "erk03 - ERK Multi-KIN" is required, if the kernel shall manage several kinematics simultaneously. The robot libraries "opk*" from ABB, Kuka, Fanuc, Motoman, Stäubli und PKM are optionally available like in EASY-ROB™.

The Motion Planner requires the basic module "ERK Motion Planner". This allows moving kinematics with the motion types PTP, SLEW, LIN and CIRC with respect to the robot base.

The Motion Planner can be expanded with the Motion Planner options "opm*". The option "opm01" for example is required for a tool leading movement.

For the integration into your own application we are pleased to offer you our support.

Item No.	Product

ERK Basic Moduls

erk01	ERK Kinematics - Single-Robot
erk02	ERK Motion Planner - PTP, LIN, CIRC - Workpiece leading - 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 Tool leading
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

New Robot Jog Mode „TCP Base“

The robot TCP can be jogged now with reference to his robot base. This corresponds - including the „TCP Tool“, „TCP World“ and „Robot Joints“ modes – to every manual moving type found on the programming device of the robot control.

Jogs the TCP in world coordinates

NEW: Jogs the TCP with respect to his robot base

Jogs the TCP with respect to world coordinates

Jogs the robot axes

NEW: Zoom on

- World ()
- 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

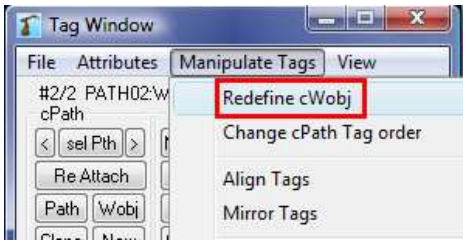
Manipulation of paths and tags

New possibilities to manipulate paths and tags have been added. For example: Mirroring of paths, alignment of tag points to an axis, partial copy of tag-positions, reversion of path-directions and redefining of work-objects. Additionally the length, angle and distance to previous and following tags can be displayed.

External axe-values (of e.g. a positioner or a rotary tilting table) can be calculated automatically in dependence of the path-length by using “AutoCalc“.

Redefine cWobj Position

The use of “Redefeine cWobj” changes the work object-position, without changing the position of the tag points. Use this function if you have difficulties to place tags or plan whole paths because of the unfavorable position of the work object.



Open the Tag Window an chose *Manipulate Tags > Redefine cWobj*

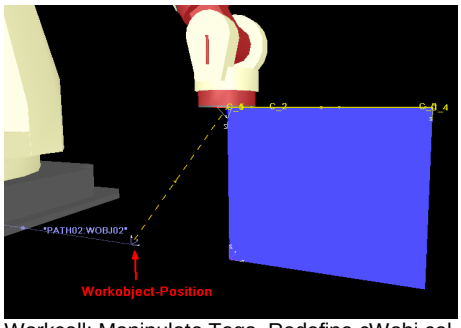
The Frame Dialog opens. There you can enter the new work object-position.

A simple example will explain the advantage of the function “Redefine cWobj”.

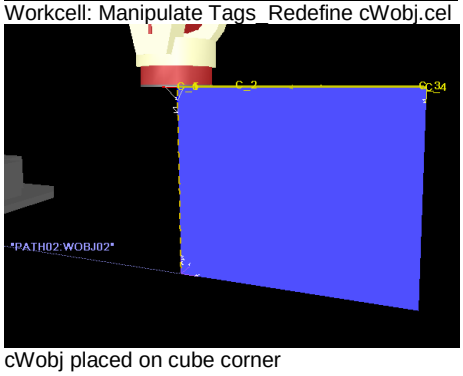
The work object position is outside of the blue work object. Reference system for the tag points on the work object is the cWobj. This unfavorable position of the cWobj makes it more difficult to attach and position new tag points at desired positions of the work object. The reason for that is simple: the reference system is outside of the work piece and not e.g. the coordinate origin of the geometry.

The planning of tags on the geometry will be relieved, if the origin of the cWobj will be placed on one of the eight cube corners.

Another typical application for this function is the use after reattaching paths.



Workcell: Manipulate Tags_Redefine cWobj.cel

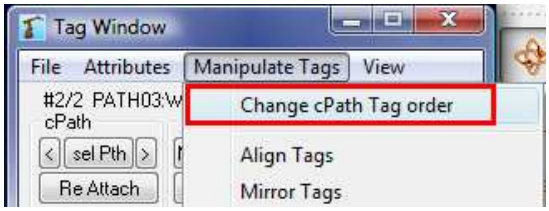


cWobj placed on cube corner

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

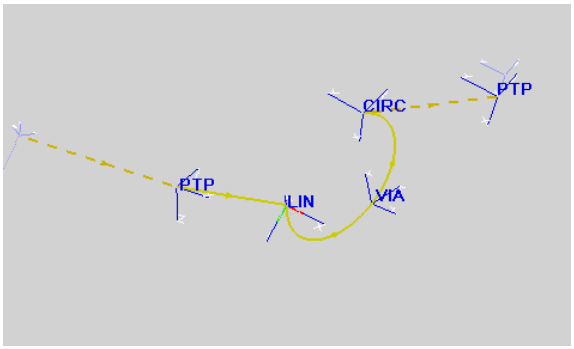
Change cPath Tag order

If you change the approach-direction of a given tag-order (e.g. reversion of a path direction) the motion type (PTP, LIN, VIA, CIRC) which is used for the movement to every specific tag point, has to be changed.



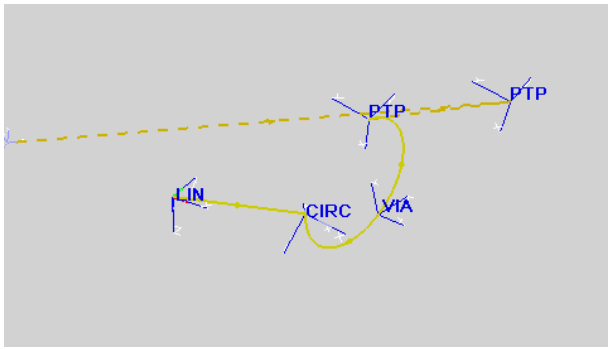
The approach direction of a tag sequence and the associated motion types can be changed by selecting *Manipulate Tags* > *Change cPath Tag order.*

The following sequence of motion types will illustrate the problem.



Manipulate Tags_Change Tag Order.cel

Original path: **PTP → LIN → VIA → CIRC → PTP**



Clone the original path previously!!!

New path: **PTP-NEW → PTP → VIA → CIRC → LIN**

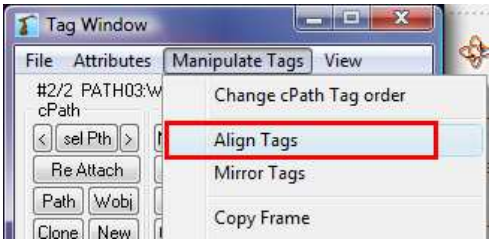
The use of “Change cPath Tag order” deletes the first tag by moving it out and adds a new tag with the PTP motion type as last tag point in the sequence.

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 – alignment of tag points

The function “Align Tags” will align the x-, y- or z-axis of a chosen tag to an axis of a selected reference tag.

This serves to reduce the axial movements of a robot, which results in a higher precision of the motion sequence and a higher quality.

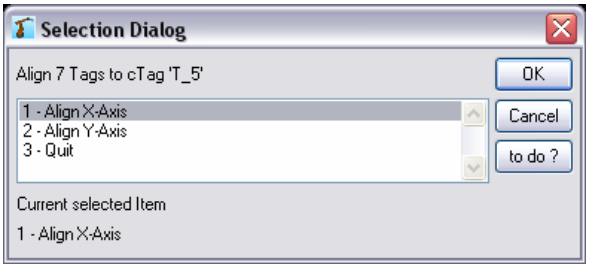


Select *Manipulate Tags* > *Align Tag* to align the chosen tags.

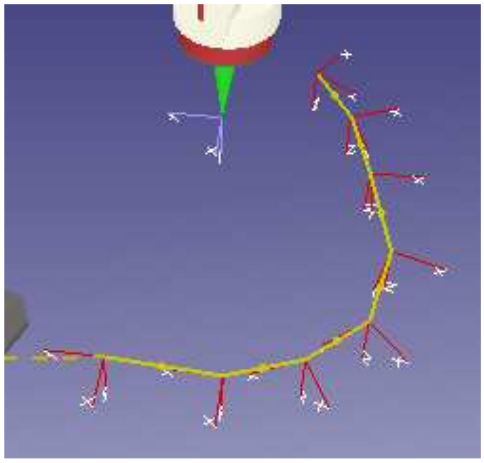
Note:
By selecting single tags you can decide which tag will function as reference tag and which tags will be aligned to this reference tag.

The order of selection is important: The last tag you select from the list will be set as reference tag. All other tags will be aligned to this tag.

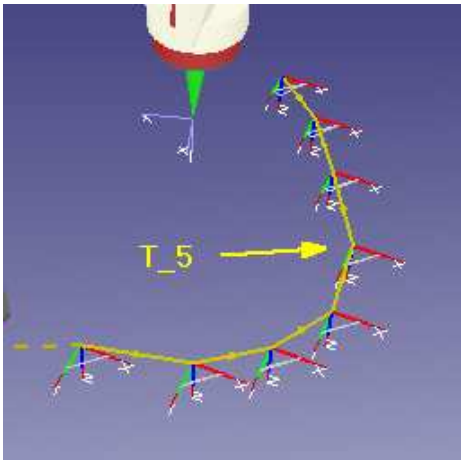
Finally you can choose which axis of the tags should be aligned.



Only the x- and y-axis are available, because the z-axis has been chosen as approach axis.
Menu → Attributes → Tag Approach Direction



Manipulate Tags_Align_Tags.cel

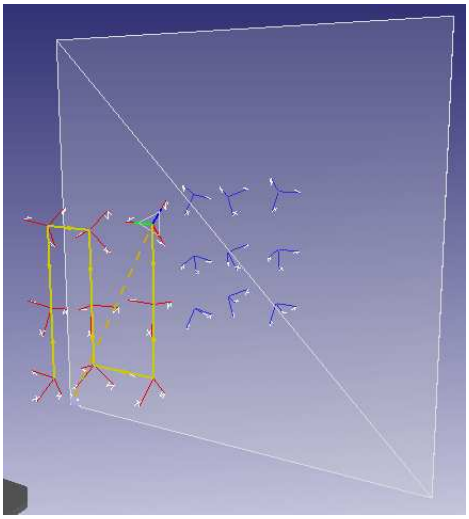


The x-axis of the tags has been aligned to the x-axis of the reference tag “T_5”.

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 – mirroring of tag points

It is possible to mirror one or several tags using a mirror plane.



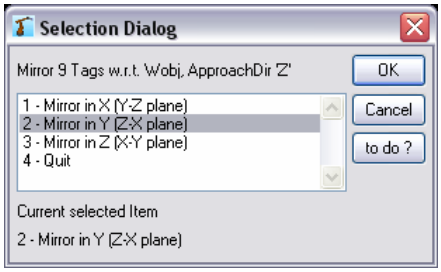
Work cell: Manipulate Tags_Mirror Tags.cel

The reference system is set by the cWobj-coordinate system.
As you can see: The tags you want to mirror are red. The mirrored tags are blue. The white plane is the mirror plane.

Open the Tag Window and select the tags of the path you want to mirror.

Hint: Clone the path previously

Select *Manipulate Tags* > *MirrorTags* to mirror the selected tags.



Before it is possible to mirror tags the mirror plane has to be set first

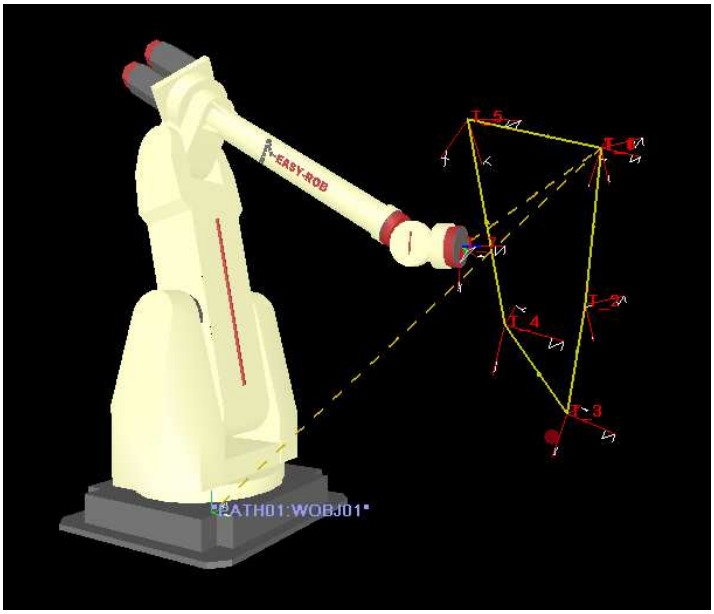
Mirror in X *(mirror plane put up by Y-Z-axis)*
Mirror in Y *(mirror plane put up by Z-X-axis)*
Mirror in Z *(mirror plane put up by X-Y-axis)*

Note: The approach direction plays an essential role. The approach axis (in this case the z-axis) is dominant and will be overtaken. It decides how the mirrored coordinate system will be aligned. If the z-axis is chosen as approach axis the z- and x-axis will be mirrored and the y-axis will change its direction. If this is not desired, you have to rotate every mirrored tag by 180° about the z-axis.

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

Copying tag point-position/ -orientation

Every tag position (Tag-Frame) consists of the x-, y- and z-positions and of an orientation part. Single parts of this reference tag can be transferred to selected tag points.



Manipulate Tags_Copy Tags.cel

You can chose between 6 different copy commands:

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

The procedure for copying tags is always the same:

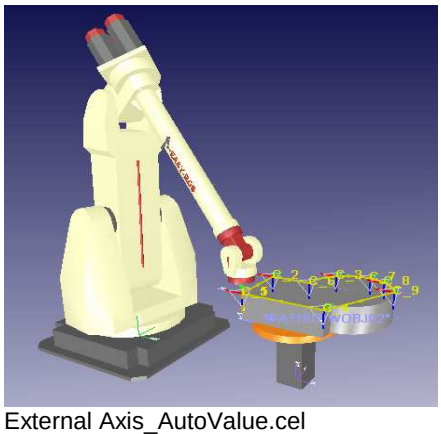
1. Open the Tag-Window with a double click on 

2. First you have to choose which tags have to be aligned to the position and orientation of the reference tag. The order of choosing the tags is important. The last tag selected from the list will be set as **reference tag point**.
3. Select *Manipulate Tags > Copy (Frame; Orientation; Positions; X; Y; Z)* to align the selected tags to the reference tag.
4. Check and apply your settings in the following dialogue. The settings will be applied with a click on “OK”.

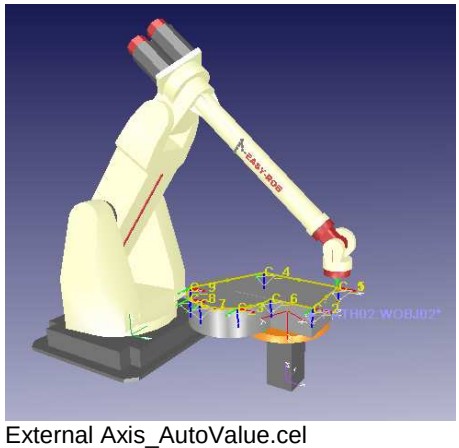
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

Automatic calculation of external axis values in dependence of the path length

The values for external axes are saved within the tag data. These values have to be determined, according to their purpose, so that e.g. the position at the work object can be reached in every situation.
Using the new function “Axis-Value-Auto” the external axis values will be calculated automatically in dependence of the path length.
The example „External “Axis_AutoValue.cel“ shows how the turn-table is steered by the 7th external axis of the robot. (See screenshots below)



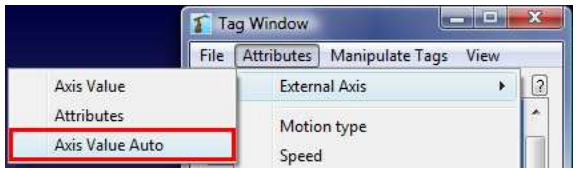
External Axis_AutoValue.cel



External Axis_AutoValue.cel

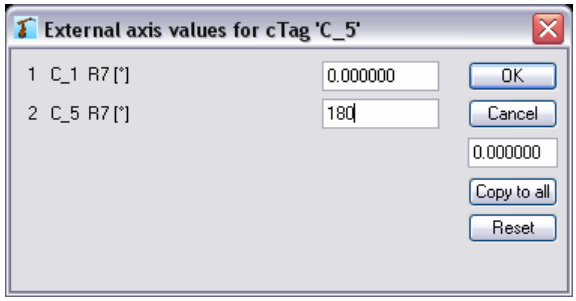
An equal rotation of the turn table in dependence of the path section can be set by using the function **“Axis Value Auto“.**

You can find the function in the Tag-Window Menu *Attributes > External Axis > Axis Value Auto*



Chose a tag or a path in the Tag Window with a click on *Attributes > External Axis > Axis Value Auto*. Enter a start and an end value.

The specified rotation will now be distributed equally over the whole path section. This distribution depends on the distance between the single tag points and the set motion types.

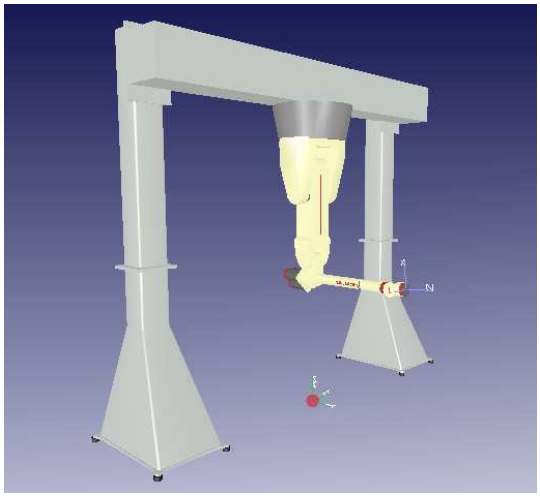


EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

New kinematics type: JET Robot (gantry robot)

JET robots with the ID 127 (respectively. 128 for A2A3 coupling) using the serial structure TyRyy:Rxyx or TyRyy:Rzyz with 4 configurations are supported and can be created by customers (e.g.. KR 30 JET, KR 60 JET).

Two- and three-axis gantry kinematics (ID 133) can be created in the following combinations:
Txyz, Tyxz, Tzxy, Tzyx, Txzy, Tyzx bzw. Txz, Tyz, Tzx, Tzy, Txy, Tyx



er431-jet.rob

Example of the KR 30 JET

On the following page you will find a list of robot kinematics, which are supported by EASY-ROB™.

Inverse Kinematics ID

The „Inverse Kinematics ID“defines the mathematical solution for the reverse transformation (inverse kinematic) and forward transformation for each robot. EASY-ROB™ offers direct solutions for different kinematics, like 3- or 5-axis gantry robots, articulated robots, Scaras, etc.

Every Kin-ID has its own Sub-ID (default 0)

The Kin-ID and the Sub-ID can be edited in the Robotics Menu:
cRobot Kinematics -> Kinematics Data -> Inverse Kinematics ID -> Special Inverse Kinematics

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 table

Kin-ID	Name	Sub ID	Kin_Type	Comment
0				no inverse kinematics available
1	DLL #1			User-defined kinematics in „er_kin.dll“ #1
2	DLL #2			User-defined kinematics in „er_kin.dll“ #2
3	DLL #3			User-defined kinematics in „er_kin.dll“ #3
4	DLL #4			User-defined kinematics in „er_kin.dll“ #4
5	DLL #5			User-defined kinematics in „er_kin.dll“ #5
6	DLL #6			User-defined kinematics in „er_kin.dll“ #6
7	DLL #7			User-defined kinematics in „er_kin.dll“ #7
8	DLL #8			User-defined kinematics in „er_kin.dll“ #8
9	DLL #9			User-defined kinematics in „er_kin.dll“ #9
10	DLL #10			User-defined kinematics in „er_kin.dll“ #10
11	DLL #11			User-defined kinematics in „er_kin.dll“ #11
12	DLL #12			User-defined kinematics in „er_kin.dll“ #12
13-99	User Inverse Kinematics			User-defined kinematics in „er_kin.dll“ #13-99
100	NumSol	0	any	Numerical solution kinematics >= 6 axes Further parameters: Tolerances, Joint Weight, Mask Vector
100	NumSol	1	any	Numerical solution kinematics with less than 6 axes (Agreement Approach axis)
110	Articulated		RzRyy:Rxyx , RzRyy:Rzyz	and Tracking axis „Standard RRR:RRR on Y-Track“
111	Articulated		RzRyy:Rxyx , RzRyy:Rzyz	like 110, with „Backlink“ respectively. „A2A3 coupling“ „Back Link RRR:RRR on Y-Track“
116	Articulated			like 110, solution w.r.t Robot Base
117	Articulated			like 111, solution 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 table

Kin-ID	Name	Sub ID	Kin_Type	Comment
120	b+m			T1 Painting robot
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			Like 127, with „Backlink“ respectively. „A2A3 coupling“
131	SCARA	0	RzRzTzRz	
133	Gantry 2 axes	13,23	Txz,Tyz, Tzx, Tzy, Txy, Tyx	2 axis gantry
	Gantry 3 axes	123,0	Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx	3 axis gantry
	Gantry 1 axis	1,2,3	Tx, Ty, Tz	1 axis gantry, Conveyor
134	Gantry 2+1 axes	13, 23	Txz,Tyz, Tzx, Tzy, Txy, Tyx, Rz	2 axis gantry + Rz rotation axis
	Gantry 3+1 axes	123, 0	Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, Rz	3 axis gantry + Rz rotation axis
135	Gantry, 3+2 axes		Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, CA=Rzx	5 axis gantry with C- und A-axis
136	Gantry 6 axes		Txyz,Tyxz,Tz xy,Tzyx,Txzy, Tyzx, Rzxz,Rzyz	6 axis gantry with Rzxz or Rzyz
114	Abb			Optional
118	Motoman			Optional
115	Staubli			Optional
112	Kuka			Optional
113	Fanuc			Optional
132	Tricept			Optional
137	PKM			Optional, Delta-kinematics, FlexPicker

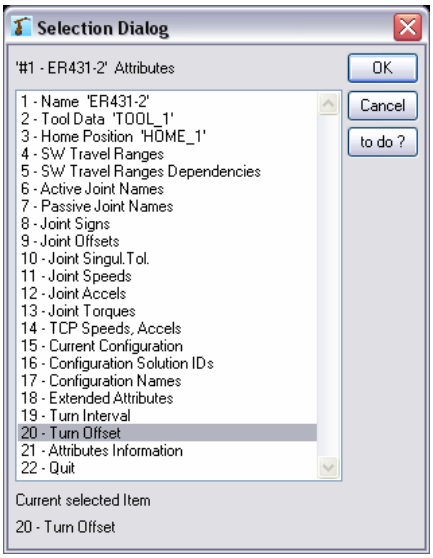
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

New robot-attributes

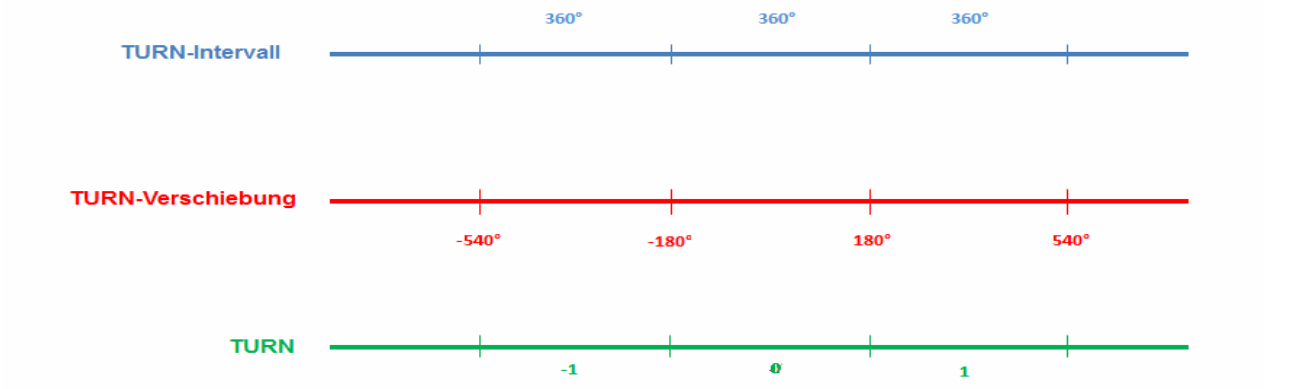
Active or passive robot joints, kinematics or devices can be named now, e.g. „vertical axis" or „transverse drive" instead of „Joint_1". Besides, names can be assigned to configurations, like e.g. "KUKA S'B010" or "Fanuc NUT" instead of "Config_1".
In addition to the turn-interval, turn-offsets for every joint can be defined. This is necessary for the proper synchronization with the robot-controller.

Turn-Offset

Open the Kinematics Window and click on Attributes



The graphics below will explain the Turn-Offset more detailed:



Using a Turn-Interval of 360° and a Turn-Offset of -180° results in a Turn = 0 if q is in]-180,180], = 1 in]180,540] and = -1 in the range of]-540,-180].

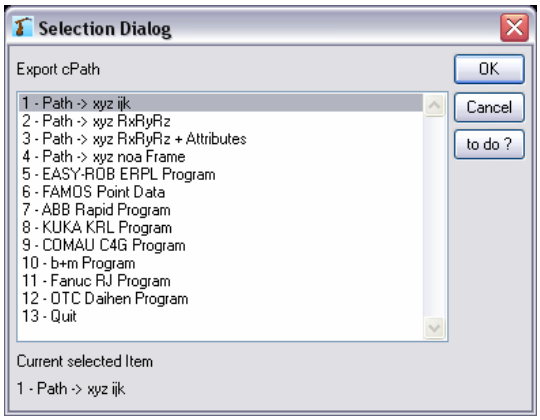
EASY-ROB™ Stefan Anton Hans-Thoma-Str.26a 60596 Frankfurt/Main Germany Tel. +49 69 677 24-287 Fax. +49 69 677 24-320 info.er@easy-rob.com www.easy-rob.com

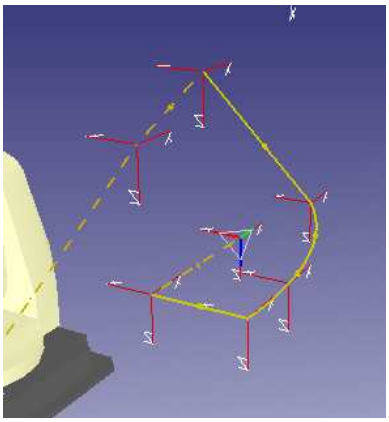
Exporting paths into native robot programs

The generic export of paths into native robot programs has been enhanced for the control types of ABB, KUKA, COMAU, Fanuc, b+m and OTC. Further control types can be integrated on our customers wish.

The Work cell „Path-Export.cel“ serves as an example. The path has 7 tags, which has been exported to native robot programs. Open the Tag Window and select Menu File -> Save -> Export cPath.

Export Path allows you to take over paths – generated with EASY-ROB™ - into robot programs in a syntactically correct way. This is also called OLP sometimes!? For this kind of application, we recommend to use Famos robotic®, an offline programming tool for robots and process-technology.




Path with 7 tags

Following some export examples:

Export Path ABB

Output-file 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

Output-file 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

Output-file 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

Output-file 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

Collision

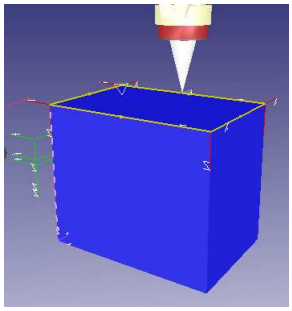
The new collision-algorithm „PQP” allows you to define tolerances. That means that collision will be indicated if two bodies approach each other and reach a minimal distance.

This tolerance can be set individually for **each body**.

Geometry-specific tolerance

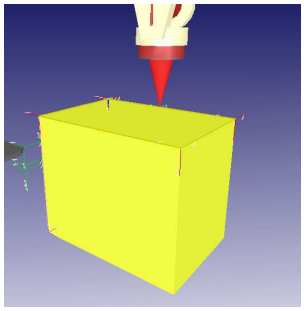
Example: Collision-Tolerance.cel

In this example the collision-tolerances varies by using ERPL-commands in the program.



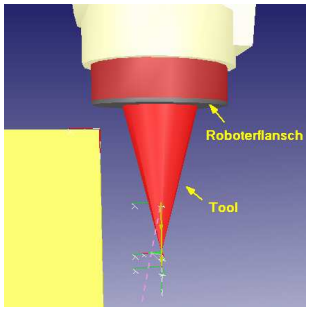
Collision-Tolerance.cel, first without collision

In this example the robot moves along the cube with a collsion-tolerance of 0mm. As you can see, there is no collision.



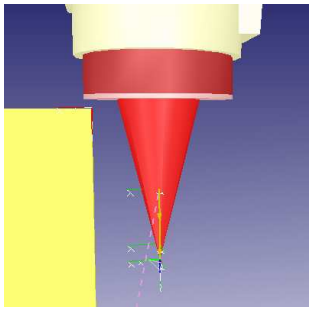
Collision-Tolerance.cel, with collision

In the second example a global collision value of 5 mm is set. Now, collision will be shown.



A collision-tolerance of 50mm is set for the tool geometry and a collision-tolerance of 20mm for the robot-tip.

Result:
The tool collides with the cube, because the distance is lower than 50mm. The tip does not collide because the distance is more than 20mm.



Robot is moving a little bit further

Result:
The tool is colliding furthermore. Now also the tip shows collision, because the distance has been reduced to less than 20mm.

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

New ERPL and ERCL commands

MSG Text

“Text“ will be shown in the Program Window

NATIVE Native-Text

The native command can be used directly in the Post-Processor API.
It has no influence on the simulation.

ERC TURN_INTERVAL Ax1 ...Axn [deg]

TURN-Interval for each axis Ax1…Axn, in the range between [0°, ∞°]

ERC TURN_OFFSET Ax1 ...Axn [deg]

TURN-Offset for each axis Ax1…Axn, in the range between]-∞°, ∞°[

ERC GRAB_TO DEVICE devname targetdevname

The device with the name ’devname’ will be grabbed by the device with the name ‘targetdevname’.

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, Method-Class ER_CAPI

Many new API functions for individual product customization and special solutions have been added.
These new functions allow to control EASY-ROB™ out of an own application or to exchange data in a bidirectional way.
The method-class ER_CAPI serves as interface for the EASY-ROB™ Multi-Program and EASY-ROB™ DLL version as well as for the extensions API-INV, API-IPO, API-DYN, API-UserDLL, API-PostProc and API-Sensors.

The exported class ER_CAPI structures every EASY-ROB™ API-function and simplifies the use. The class ER_CAPI is a pure method-class, which is defined in the header-files „./er_dvlp/**er_capi.h**" and „./er_dvlp/**er_capi_types.h**". All methods are standard ANSI C. „Old" ANSI C functions, which are defined in the header-files „./er_dvlp/**er_dvlp.h**" and „./er_dvlp/**er_dvlp_ext.h**", are still available because of the compatibility.

AUX_UPDATE_IDX_SET_FOCUS
ER_CAPI_SYS_MATHEMATICS::circ_center_point() considers additionally the angle until the VIA-point.

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

Other Improvements

- Teach-Window with new dialogue and all motion commands
- Scalable Device Manager dialogue
- More resolutions for AVI-Recorder available
- The sensitivity and threshold value of a 3D Space Mouse can be set in the environment file
- New parser-functions allow access to kinematic robot-length an more
- Advanced Status Output for the output of e.g. axis values in every simulation step

Configuring the Space Mouse with the environment file

Bild: 3DConnexion

The environment file “easy-rob.env” (Alt+Shift+E) allows you to customize basic functions of your Space Mouse:

```
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	Activates/deactivates Space Mouse Menu
S3DM_SPEED	1	Sets sensitivity of the Space Mouse
S3DM_THRESHOLD	1	Configures the threshold value of the Space Mouse. The set value determines how far the mouse has to be moved until it responds.

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

Contact

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 Customer area

Online available: Program Updates and Robot libraries

Web: www.easy-rob.com/special/customer-area

Access data:

User: customer
Password: **********

Notes

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