The ROBOTICS Cycle Time Analyzer.
The first "time".

WPA Nr : 1193.
M.C. Willems.

In order of : TUE-WPA

Professor : Prof. Dr. Ir. A.C.H. van der Wolf

Coaches : Ing. J.J.M. Schrauwen
: F. Soers

Author : M.C. Willems

Eindhoven, 8 november 1991.

ONDERZOEKOPDRACHT

TECHNISCHE UNIVERSITEIT EINDHOVEN 28 juni 1991
Faculteit Werktuigbouwkunde
Vakgroep WPA

Student	: J.P. Melio, M.C. Willems
Hoogleraar	: Prof.dr.ir. A.C.H. van der Wolf
Begeleiders	: Ing. J.J.M. Schrauwen, F.G.J. Soers
Start	: Juni 1991
Einde	: Augustus 1991
Titel	: De CTA module van "Robotics"

Onderwerp
Binnen het Robotica pakket Robotics van McDonnell Douglas is een module CTA (Cycle Time Analysis) aanwezig. Deze module wordt gebruikt om werkelijke snelheden en versnellingen aan het model toe te voegen, zodat off-line programmeren nauwkeuriger wordt.
Binnen de CTA module kan de cyclustijd op twee manieren worden aangepast:

- Door het terugkoppelen van de werkelijke snelheden en versnellingen verkregen door metingen aan de robot.
- Snelheden en versnellingen worden geschat aan de hand van technische specificaties van de fabrikant.

Opdracht
Onderzoek de bruikbaarheid van deze module en schrijf een beknopte handleiding. Bij het onderzoek wordt gebruik gemaakt van de bestaande configuratie van de gemodelleerde Kuka-cell.

Prof.dr.ir. A.C.H. van der Wolf   Ing. J.J.M. Schrauwen   F.G.J. Soers

SUMMARY.

Off-line programming of production machines is becoming increasingly important nowadays. Many software programs are developed for off-line programming. It is important that the software models of the production machines, which you are modeling, imitate the real production machines very accurately. ROBOTICS is such an off-line program package and this program has a module, Cycle Time Analyzer, for the dynamic calibration of a robot.

With CTA the total work area, for every axis, for the whole speed range of a robot is examined and stored in a file. The file is then connected to the robot in the software package, to predict accurate cycle times during a simulation.

The following actions are necessary to run the Cycle Time Analyzer.

- Write an options file. This file is the basis of your test. It contains the initial positions of the robot for every axes and the number and lengths of the test moves. It also contains the initial position of the robot, the number and length of the straight line movement.

- Run CTA on the HP workstation. CTA creates a cell and seven sequences: six sequences for the six different axes and one for the straight line move. The cell contains the robot (= a device) and the tpoints for the straight line move. These sequences perform the moves which are tested.

- Write an USR-file. The USR-file is the skeleton of your robot program. It contains the commands to turn a signal line "ON" and "OFF". This signal is used to determine the time of a test move. It also contains the commands of a loop to automatically cycle through the different robot speeds, during the tests.

- Run COMMAND on the HP workstation. In COMMAND, the USR-files and the sequences are processed into CSP-files.

- Run COMMAND on the VAX. In COMMAND on the VAX, the CSP-files are postprocessed into robot programs (SRC-files). These are in german. Translate them to dutch in an editor on the VAX. Change, if necessary, the $WISTAT commands.

- Download the SRC-files. Download the dutch SRC-files to the Robot Controller, via a communications program and ethernet. Use DNC to send them into the Robot Controller. The names of the programs which are send to the Robot Controller must exist of the characters HP and a two digit number.

- Run CTA on the PC. The CTA-PC module will cycle you through the determination of the timing data of all axes and the straight line movement and will produce the timing file (TIM-file).

- Transfer the TIM-file back to the workstation. Place the TIM file is the system library, in order for all users to access it.

Running CTA in it self was and is not a big problem. But many small problems had to be sorted out before the actual CTA was performed. This was due to the fact that it was the first "time".

PREFACE.

On 28 june 1991 I got the research assignment: examine the McDonnell Douglas ROBOTICS Cycle Time Analyzer and write a simple user guide for our situation. Do this for the existing KUKA workcell model.

This report does not intend to replace the CTA user guide but should rather be used as an supplementary to the CTA manual for the computer configuration existing on the TUE.

I want to thank my coaches and especially Henk van Rooij, who helped a lot with the small computer problems that occurred and Anton Smals, who helped with the robot and Eric Nicole for his long distance help.

CONTENTS

Chapter 1. INTRODUCTION.

Off-line programming of production machines is becoming increasingly important nowadays. More and more off-line programming software packages are developed. With these software packages you can make a model of your production machine. It is very important that this software model imitates the real production machine very accurately. ROBOTICS is such a off-line programming software package and CTA is a module of ROBOTICS for dynamic calibration of a robot model. The module determines the accelerations and velocities for the whole work area of a real robot and stores these in a data file. This data file is then connected to the robot model in the software package.

In this report, the Cycle Time Analyzer (CTA) [2][3] is examined. It is used to predict accurate cycle times for the KUKA robot workcell model(FALC [5]). A shortcoming of this cycle times determination is that the position of the robot during the movement tests can be chosen. It is difficult to say whether you determined the robot's cycle times in the right position of the robot. Another shortcoming is the welding thread support post. This post is blocking the movement of the first axis from 0 degree to 160 degrees. So only half of the total range of the first axis is used for the movement test.

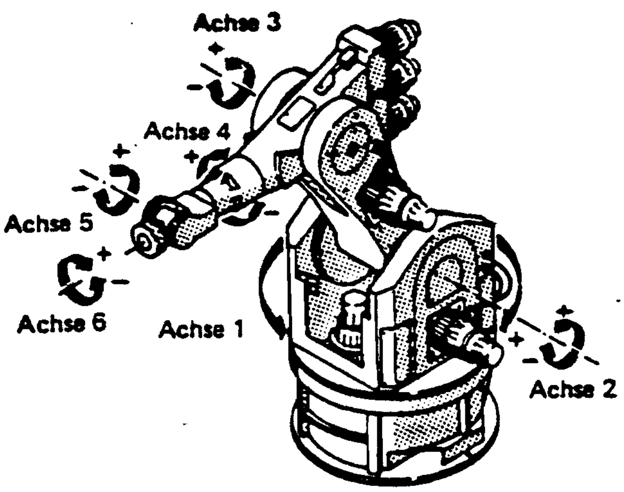


Figure 1.1. The KUKA-robot with the different axes.

ROBOTICS is a software package designed by McDonnell Douglas for off-line programming of Robots and Robot cells. The package consists of five modules :
BUILD, PLACE, COMMAND, ADJUST and CTA. With BUILD [1] you can build a device (robot or manipulator etc.) consisting of separate parts, which are modelled in UNIGRAPHICS. In PLACE you place the devices together in a cell and in PLACE you can also simulate moves and operations. When you have made moves and/or operations which you want to use for the real robot, you save them in a sequence. In COMMAND [4] you can translate these sequences into SRCL-language, which is the language that is used by the real robot. In ADJUST you calibrate the dimensions and the position of the modelled robot and other devices. In CTA [2][3] you calibrate the speeds and accelerations of the modelled robot.

The ROBOTICS-module Cycle Time Analyzer is a software package that aids PLACE in accurately predicting robot work cell cycle times. CTA is based on two methods :

- Empirical data collection method. With this method, timing data is collected for a representative set of robot motions of an actual robot. The robot is programed to make a series of movements, from very small to very large. These movements are timed by a connected PC. After recording this timing data, CTA processes them, to create a timing file for that robot. This timing file is then connected to the robot in the ROBOTICS package.

- Manual timing file generation method. This is a method where the required robot data (not timing data) is entered in response to a series of prompts, eliminating the need of collecting timing data from a set of robot motions. An actual robot is not required for this method. The timing file is as accurate as the data, which is entered. Perhaps this method is interesting for pre-examining a certain robot.

McDonnell Douglas claims that in most cases the predicted cycle times, using these two methods, are accurate within 5% of the observed time. This prediction also goes for those cases where other moves and different loading conditions were used, than the moves and loading conditions during the data collection. This is due to the interpolation and the extrapolation algorithms which are used to generate accurate cycle time predictions.

The interpolation algorithm is used when a motion falls within the range of the tested motions. The extrapolation algorithm is used to predict cycle times beyond the maximum or minimum limits of the tested motions and/or speeds, but such extrapolated values are often less reliable than the interpolated values. Of course the overall accuracy is always a function of the amount of timing data (number of test moves). Its advisable to use a range as big as possible (from 1 degree to the joint constrains for each joint).

CTA consists of a software module that runs on a workstation and another software module that runs on an IBM-PC or compatible. A hardware Robot/PC interface is used to connect the Robot Controller I/O-ports to the PC. The PC is used to time the robot motions on the shop floor. The interface converts the Robot Controller output signal to an interrupt which can be received by the PC. The PC times the moves during the test motions of the robot. When all the axes are tested, the separate timing data is processed into a timing file (TIM-file). The processed timing data are then transferred from the PC to the workstation using a communications package. Once located in the proper directory on the workstation it can be used by PLACE to predict accurate cycle times for that robot.

Each time, during a PLACE-session, a device is merged into a cell, PLACE searches for a TIM-file with the same name as the robot's DCI-file. If a TIM-file is found, its cycle time model is used whenever that device is moved. Whenever a TIM-file is being used the symbol " ↑ " appears after the device name in the joints display window.

Remark !!

To use CTA it is necessary to have some experience with the Robot Controller and with PLACE.

Chapter 2. MANUAL TIMING FILE GENERATION METHOD.

With this method the user is able to create a timing file without requiring an actual robot. The user must enter the desired data in response to a series of prompts :

ENTER DEVICE NAME :
MINIMUM / MAXIMUM PROGRAM SPEED FOR JOINT MOTION :
MINIMUM / MAXIMUM PROGRAM SPEED FOR STRAIGHT LINE MOTION :
NAME OF JOINT CRD FILE :
NAME OF STRAIGHT LINE CRD FILE :
MAXIMUM JOINT VELOCITY FOR JOINT 1 :
MAXIMUM JOINT VELOCITY FOR JOINT 2 :
MAXIMUM JOINT VELOCITY FOR JOINT 3 :
MAXIMUM JOINT VELOCITY FOR JOINT 4 :
MAXIMUM JOINT VELOCITY FOR JOINT 5 :
MAXIMUM JOINT VELOCITY FOR JOINT 6 :
MAXIMUM JOINT ACCELERATION FOR JOINT 1 :
MAXIMUM JOINT ACCELERATION FOR JOINT 2 :
MAXIMUM JOINT ACCELERATION FOR JOINT 3 :
MAXIMUM JOINT ACCELERATION FOR JOINT 4 :
MAXIMUM JOINT ACCELERATION FOR JOINT 5 :
MAXIMUM JOINT ACCELERATION FOR JOINT 6 :
MAXIMUM STRAIGHT LINE VELOCITY :
MAXIMUM STRAIGHT LINE ACCELERATION :

It is necessary to have these data available. Most of these data are contained in the robot user manual. (if you do not have an actual robot, you probably do not have a robot manual?!) Perhaps it is difficult to obtain these data without a robot manual. The data entered into CTA with this method can also be defined in BUILD. For this reason, you can almost always use BUILD instead of this option. The main difference is that BUILD does not work in robot speed units. The manual timing file generation method also gives you a way of changing timing parameters during a PLACE-session.

This method is not further investigated, because of the reasons mentioned above and because we do have an actual robot. Perhaps this method is interesting for pre-examining a robot. If you do not have an actual robot but want to know whether the robot, which you are interested in, can handle the wanted tasks, you run this manual timing file generation and use the timing file in a PLACE simulation.

Chapter 3. EMPIRICAL DATA COLLECTION METHOD.

Using the Empirical Data Collection Method of CTA, you can generate timing parameters from a physical robot and use these parameters in PLACE. CTA requires you to make an options file (OPT-file). This OPT-file is the basis of your test. It contains the position of the robot before doing the test moves (this is called the initial position) and it contains how many moves the robot must make and how big (angles) these moves are. CTA then creates a cell and a set of sequences that perform the moves. This cell contains the device which is tested and it contains a series of tpoints for the straight line movement. The sequences which CTA creates are, for each axis, the movements of the robot. The robot will move from its starting position with increasing angles until the moves are so large that they will exceed axis constraints. These sequences are translated with COMMAND into a set of robot programs. A program for each joint which contains the different moves and a loop to cycle through the different speeds.

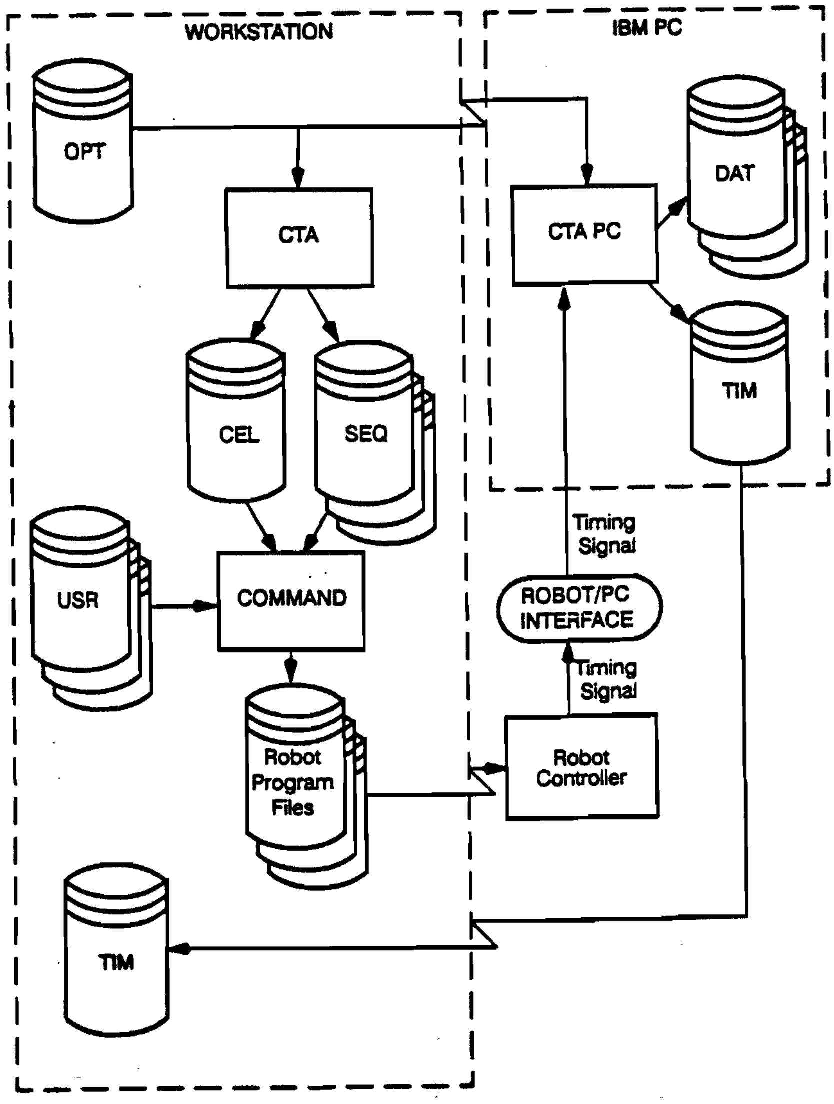


Figure 3.1. CTA components and organization.

The following steps are required to generate a timing file.

3.1. Setting up an options file.

The main purpose of the options file (filename.OPT) is to define how the robot will move while the timing data are being collected. The order of the data must always be the same. The options file (as I defined it) for the KUKA is shown.

```
DEVICE NAME = KUKA
DCI NAME = KUKA
JOINT CRD NAME = JOINTSM
STRAIGHT CRD NAME = KUKACART
OUTPUT NAME = TIJSJ
CELL OR DEVICE = CELL
CELL NAME = TIJS
NUMBER OF JOINTS = 6
NUMBER OF JOINT SPEEDS = 10
MIN JOINT SPEED = 0.00
MAX JOINT SPEED = 100.0
JOINT SPEED FACTOR = 10.0
JOINT SPEED UNITS = PCNT
MIN STRAIGHT SPEED = 0.0
MAX STRAIGHT SPEED = 100.0
STRAIGHT SPEED FACTOR = 10.0
CONVERSION FACTOR = 1.0
STRAIGHT SPEED UNITS = M/MIN
FOR LOOP = TRUE
DATA = -79.0   60.0  -56.0   10.0   10.0  125.0   0.0 20  1.0 14 10.0
DATA = -85.0   45.0  -40.0    0.0  -54.0  226.0   0.0 20  1.0 10 10.0
DATA = -85.0   40.0 -130.0    0.0  -53.0  226.0   0.0 20  1.0 22 10.0
DATA = -85.0   48.0  -50.0    0.0   10.0  134.0   0.0 20  1.0 32 10.0
DATA = -85.0   48.0  -50.0  180.0    2.0   44.0   0.0 20  1.0 20 10.0
DATA = -85.0   48.0  -50.0  180.0   10.0  -45.0   0.0 20  1.0 34 10.0
DATA = 320.1 -731.3 1216.2   36.4  -71.9   80.6   0.0 20 20.0 20 70.0
```

Figure 3.2. The options file.

* DEVICE NAME = <name>
This is the file name of the robot, that is in PLACE, which you want to test. The robot is a device and has the name FILENAME.DEV (see Appendix 1.). You can find it in the directory where all your ROBOTICS files are. Type all filenames, that you must enter, without extensions.
* DCI NAME = <name>
This is the name of the Device Control Information (DCI, see Appendix 1.) file for the robot. This file often has the same name as the device name.
* JOINT CRD NAME = <name>
This is the name of the joint Coordinate System (CRD, see Appendix 1.) file. This file defines the relationships between the joint angles and the angles to which the speeds and accelerations apply for joint interpolated motion (= how to reach a certain point in space given joint angles). The name of the joint CRD-file must be the same as the joint CRD-filename used in the BUILD (BLD) file and in the DCI-file of that device.
* STRAIGHT CRD NAME = <name>
This is the name of the CRD-file (see Appendix 1.) which defines the cartesian position of the robot. This name must be the same as used in the BLD-file and the CDI-file.
* OUTPUT NAME = <name>
This is the name used as the prefix for all sequences and cells (see Appendix 4. and 5.) produced by CTA. The sequence names consist of the named prefix and a single digit for each joint and in case of the

straight line motion the prefix and the character "s". The name of the cell made by CTA will have the prefix and the character "s" followed by the extension CEL.
* CELL OR DEVICE = <CELL/DEVICE>
If the entry is cell, it means that a cell (see Appendix 1.) containing a robot device has already been defined (in PLACE). If the entry is device , it means that a cell containing this robot must be created by CTA, before the timing sequences can be produced.
* CELL NAME = <name>
If the previous entry is cell, then the name of the existing cell must be entered. If the entry was device this field is ignored.
* NUMBER OF JOINTS = < n >
The number of joints of the robot. See the BLD-file (see Appendix 1.), CRD-file or the DEV-file that you use for this number. The maximum is 7.
* NUMBER OF SPEEDS = < n >
The test sequences that CTA makes, will be tested at different speeds. Here you must enter how many speed changes you want. It is recommended to use at least ten speeds, in order to have good accuracy.
* MIN JOINT SPEED = < n >
The lowest speed, in robot units to be used for joint moves, usually zero.
* MAX JOINT SPEED = < n >
The highest speed, in robot units, to be used for joint moves (often 100 when the speed units are in percents).
* JOINT SPEED FACTOR = < n >
The step between the joint speeds. The range from maximum joint speed to minimum joint speed is divided by the chosen number of speeds to get the speed intervals. (max joint speed - min joint speed/number of speeds).
* JOINT SPEED UNITS = <name>
The units in which the joint speeds are expressed (PCNT, MM/SEC, INCH/SEC).
* MIN STRAIGHT SPEED = < n >
The lowest speed in robot units, to be used for straight line moves (usually zero).
* MAX STRAIGHT SPEED = < n >
The highest speed in robot units to be used for straight line moves. The highest straight line speed depends on were the straight line move is performed.
* STRAIGHT SPEED FACTOR = < n >
The step between the straight line speeds. ((max straight line speed - min straight line speed) / number of speeds).
* CONVERSION FACTOR = < n >
This is a number which, when divided by the robot's own straight line speed, converts it into inches per second. (when you use mm/sec use 1.0).
* STRAIGHT LINE SPEED UNITS = <name>
The unit in which the robots straight line speeds are expressed.
* FOR LOOP = <TRUE/FALSE>
If true then the robot's native language can be used to create a loop in each test program to automatically cycle through the entire range of speeds for that sequence of motions. If false, the operator will be prompted by CTA to manually change the robot speed and has to be rerun once for each speed.
* DATA = < n >< n >< n >< n >< n >< n >< n >< n >< n >< n >< n >
The data is divided into lines and columns.

The data lines specify the robot's starting positions for each sequence of moves. They also indicate the number and the length of the test moves. Each joint (axis) is tested with two sets of moves. The set of short moves might cover one tenth of the total range of motion of a joint. The set of long moves should cover the whole range of motion. To determine the range of motion (axis constraints, see Appendix 3.), see the BLD-file or examine it in PLACE. There must be a data line for each joint of the robot. The first data line represents the first joint, the second line the second joint, etc.. The last data line (if you have six joints, the seventh) is reserved for the straight line movement.

The columns on each data line are reserved for initial position values. The first column is for the angle of the first joint, etc.. So the third column of the second line is for the angle of the third joint in the starting position, for the testing movements for the second joint. Positions values for any joint which not exists should be set to zero (see column 7). The first seven columns together form the starting positions (called the initial position) of the robot for the test moves. The initial position of the joint that will be tested must be in the starting position near the middle of its range of motion. The other joint angles could have the value which you like them to have. They form the position of the robot during the test.

Columns number eight to eleven represent a number of moves and two move distances. The first group of moves (columns eight and nine) are the short moves. (for example you can define twenty moves increasing by one degree per step by: <20> <1.0>. In this case the joint will move in a sequence from zero (starting position) to +1, -2, +3, -4, +5, ..., +19, -20). The second group of moves (column ten and eleven) are usually the longer moves. The number of moves should be chosen so that the whole range of motion can be used and no joint limits will be exceeded. The group of moves (long and short) will be run for each speed setting!!

The last data line specifies a set of straight line moves. CTA creates a cell (outputname + s) which contains tpoints that are used as end points of the straight line moves. Again short and long moves. All the tpoints are defined by changing the X-position of the initial position from the data statement. The first seven columns represents the position and the orientation of the initial tpoint, the tpoint which will be in the middle of the testing range. Column eight until eleven are again the number and the length of the short and long moves.

Remarks !

-You better not write an options file in an editor yourself. This causes many errors which are hard to detect. Better is to copy an existing (and working !) OPT-file to your directory and edit it.

- Always use capitals (upper case characters) for the names you type. Lowercase characters will cause empty sequences.

- The straight line move will cause problems sometimes.

  The position and orientation of the tpoints are not in degrees !! The position and orientation of the tpoints given with respect to the father frame (world in this case). The values can be made visible in the move text window during a move tpoint or during a move tpoint-group simulation.

  The tpoints will be connected to the frame that is the lowest in the connection tree (closest to the world frame -> KUKA00). During the actual movement to these tpoints, the tpoints will be aligned by the tpoint of the frame that is the highest in the connection tree (-> KUKA06). The tool (TOORTS), which is moved by the robot is not a frame. TOORTS is defined in the cell as something that is connected to the last frame, KUKA06.

  The connection tree can be found in PLACE under FILE MANAGEMENT.

- The maximum straight line speed is not the same everywhere. It depends on the position, in space, of the movement. The absolute maximum is 100 meters per minute. The actual maximum straight line speed, which can be reached in a specific situation is not predictable and can only be found by trial and error. Give a certain speed and look whether or not error messages appear.

3.2. Running CTA.

Once the OPT-file has been prepared, run the Cycle Time Analyzer from the ROBOTICS menu to automatically generate the cell and sequences. CTA generates a separate sequence for each joint of the robot and one for straight line movement.

Run CTA on the workstation (the HP) and choose menu option 1:

Generate sequence and cell.

You will be prompted to enter the name of the options file (without extension). If the options file is complete and correct, messages will appear on your screen :

Reading options file.
Merging device.
Generating the sequence for joint 1.
Generating the sequence for joint 2.
Generating the sequence for joint 3.
Generating the sequence for joint 4.
Generating the sequence for joint 5.
Generating the sequence for joint 6.
Generating the sequence for straight line moves.
Generating cell.

See Appendix 4. and 5.. After generating the cell the main menu is redisplayed. Now you are ready with CTA on the workstation and can exit the main menu.

If the options file is not correct the message :

Error reading in options file.

appears and anything can be wrong. See the remarks in chapter 3.1..

3.3. Creating an User program file.

The user program file (USR-file, see Appendix 6.) is a skeleton of your robot program and therefore the skeleton of the program that will be generated by COMMAND (see Figure 3.4.). Together with the sequences generated by CTA the USR-file will form a program which contains all the moves and the speed settings. Also this program must contain statements for timing the moves.

In COMMAND, the USR-file together with the sequence (SEQ-file) will be processed into a CSP-file (Command Source Program file, see Appendix 7.). CTA creates sequences, which contain just movements so the USR-file must contain the other commands, which are necessary for the robot program. The robot program must contain a command, which turns the Robot Controller output port (nr. 30) "on" before a movement and "off" after the motion is completed. The robot program must also contain a command, which increases the speed at the end of the program, before jumping back to the beginning of the program. These commands are in the USR-file and must have KUKA syntax. This means that these commands must be in SRCL-language, otherwise they can not be postprocessed.

The USR-file must contain the commands that will turn a signal line "ON" and "OFF". The CTA PC-module uses this signal to determine when each robot move has started and is finished. Each robot move is tagged with an operation called "OUTLAY". In this sequence, a robot move, is an operation. This operation is named "OUTLAY". The contents of the operation are defined in the USR-file. You must define OUTLAY in your USR-file to turn a signal "ON" at the beginning of each move and "OFF" at the end of each move. It is recommended to add a delay of 0.5 seconds after each move, to give the PC time to record the time of the move. You can also avoid other synchronization problems by doing this.

You will need a separate USR-file for each sequence that CTA generates. All these USR-files will be identical, except for the sequence name in the &REF_SEQ statement. Here you must give the name of the sequence matching the USR-file (for the same joint).

```
FUNCTION=ON,HP91
WISTAT=T
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
&OPERATION OUTLAY
S A 30
&INC_GOTO
RS A 30
WRT Z 5
&END_OPERATION

&REF_SEQ TIJSJ1
GES ALL P1
&INC_SEG STARTUP
&INC_GOTO
&INC_SEQ TIJSJ1
ADD P1 KON+10
WRT Z 100
JMP AD 5
```

Figure 3.3. The USR-file for joint 1.

The "FUNCTION=ON,HP91" statement gives this program the name HP91. The lines "LAD P1 KON 10" to "HLT UN" are part of the for loop that automatically cycles through the set of speeds. The parameter P1 is given the value 10 (%) and as long as P1 is smaller than 100, the program can continue. At the end of the program P1 is increased with 10 (%), before jumping back to the beginning of the program. This jumping back is not really to the beginning of the program but to the address 5.

With the statement "%OPERATION OUTLAY" the definition of the operation begins. The statements "S A 30" and "RS A 30 " turns the Robot Controller output port 30 "on" and "off". The statement "GES ALL P1" sets the robot program speed to the variable P1. The "%INC_SEQ STARTUP" statement puts the robot into the correct motion mode (joint interpolated) in preparation for the initialization move and the timed moves. "STARTUP" is a program segment within each test sequence that is automatically generated by CTA.

A delay of 10.0 seconds is inserted between speed changes, to give the user time to look at the robot and PC, to see if things are working properly and the next speed setting is displayed on the KUKA control panel before continuing. When the entire speed range has been completed, the loop will be exited (HLT UN) and the program will stop.

Remark !
Which symbol, % or &, you must use in front of some commands, depends on the definition of this symbol in the CSP.DAT file. If you do not find this file (which is probably in the directory : /usr/disk2/simroot/cmd/csp/lib), you can always find out which symbol it must be by trying. The postprocessor on the VAX "wants" the symbol & in front of the commands.

3.4. Running COMMAND.

Now you have for each joint and for the straight line move an USR-file and a sequence and a cell. In COMMAND normally the USR-file, the sequence and the cell are transformed into the Robot Program File (SRL-file), the Source Robot Program (SRC-file) and the Error Message File (LIS-file). See Appendix 8.

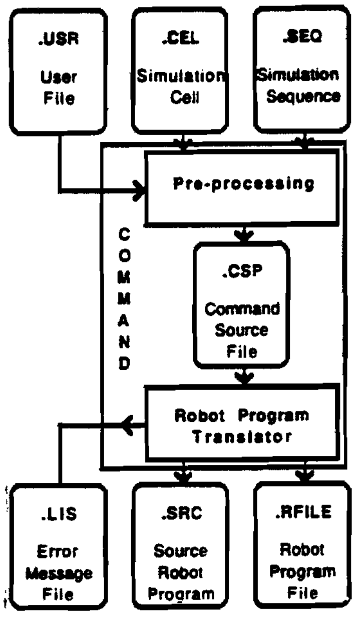


Figure 3.4. COMMAND components and organization.

However, the postprocessor is still not available for UNIX which runs on the workstation (the HP). The postprocessor runs on the VAX under VMS. Therefore, COMMAND only makes the Command Source Program file (CSP-file).

When COMMAND is started up, we choose option two :
Execute CSP.
In the next screen you will be prompted to type the name of the USR-file and the name of the CEL-file. The CSP-file is now created. Be sure that you have the COMMAND CSP Release 9.0 main menu on your screen. If this is not the case, select CSP in the option "Select Translator" on the COMMAND menu. The preprocessor has checked the CSP-file for syntax errors, but you have to check it yourself for other errors. See if all the locations, which are in your sequence, are in the CSP-file. They are at the top. See if every operation is tagged with the name OUTLAY.(see appendix 7.).

You can remove the first line :
% PREFIX_CHAR &;
This line will cause an unnecessary error during postprocessing. The postprocessor on the VAX wants the symbol & to be in front of the commands. If this is not the case, you must change The % in front of those lines into &. This can easily be done in an editor. When you have written your USR-files right, this problem will not occur. Now the CSP-file can be postprocessed on the VAX. When the postprocessor is available for the workstation, the next steps are not necessary.

3.5. Translating the CSP-file into SRCL-language.

Because the translator (postprocessor) for the KUKA robot is running only on the VAX-VMS system the translation has to be done on the VAX. All the CSP-files, six for the different joints and one for the straight line, have to be copied from the HP to the VAX. There are several ways to do this but you must be sure that they are copied to the right directory on the VAX. An old release of ROBOTICS is running on the VAX. This old release is located in a directory ROBOTICS. Postprocessing of KUKA-files is a module of ROBOTICS release 6.0.

Remark!
One way to copy the files to this directory is this one :

First you login on the VAX under "ROBOT".
userid : < ROBOT >
password : < ROBOT >
Now you are logged in to the VAX on a terminal. You are in the directory ROBOTICS ([UG_USERS.ROBOTICS]). You choose the menu option 8 -> exit, to go to DCL (VMS-level). Now you have to login on the host. You type :
$ set host tue0
You will be prompted to login again (on the host).
userid : < ROBOT >
password : < ROBOT >
Again choose number 8 from the menu to go to DCL, VMS-level. After the $ appears you type:
$ FTP
Now you are in FTP (File Transfer Package), the prompt will be :
FTP>
You type the copy command :
FTP> copy @voodoo:/users/username/filename.extension []
This means that you will copy from the HP, which is called "voodoo", from the directory which has your username, the file with the filename: filename.extension to the directory on the VAX that you are in now ([]). When you typed it right you must enter your userid and your password for the HP (don't type ROBOT here!!!). Now the file, you wanted, is in [UG_USERS.ROBOTICS] and is ready for postprocessing. To leave FTP you must type exit.

Now you have to logoff two times or you can open another window under ROBOT to get back to the login menu. In this login menu you choose option five : COMMAND. ROBOTICS release 6.0 is started. The function keys are enabled now, so you have to use the arrow keys and the spacebar. You choose option seven Postprocessing (with the arrow keys and select with the spacebar). Now choose option one : postprocessing from a Command Source Program file. Type the name of that CSP-file and give a few returns if you want the SRC-file, SRL-file and the LIS-file to have the same name as the CSP-file. Now the postprocessing starts. If there are errors or warnings, they are given at the top of the screen. Give an ENTRY COMPLETE (E) by pressing the spacebar and the ROBOTICS menu appears again. Leave ROBOTICS if you want to look in the LIS-file where the errors and the warnings are listed or postprocess another file. If the SRC, SRL and the LIS-files are correct you better copy them to your own directory on the VAX and remove them from the ROBOTICS directory. (you can do this by typing :" copy tue0$dua0:[ug_users.robotics]filename.* [] " if you are in your own directory).

3.6. Translating the German SRC-file into the Dutch SRC-file.

The robot dependent program, the SRC-file, which you have made by postprocessing a CSP-file, is in german. The KUKA robot (which is situated in the Mechanization Laboratory) works with the dutch language. You have to translate all the german KUKA commands into dutch KUKA commands (see Appendix 10). You can do this yourself or you can write a program in the VAX-editor which has the correct substitute commands (see appendix 9.). It is a very simple program due to the fact that the translation consists of simple ASCII-transformations.

3.7. Downloading the robot program (dutch SRC-file).

Downloading the robot program is only possible from the VAX, because the VAX is connected to Ethernet and is able to communicate with PC's. The actual downloading from the VAX to the PC, that is connected to the Robot Controller, is simple. The VAX is connected to the PC (that is connected to the Robot Controller). If this is not the case, in the vicinity of that PC will be another PC that is connected to Ethernet. You will use a communications program, probably PCSA, which operates under DOS. After you have logged in on the VAX (via STARTNET) you can copy the files that you need, using the normal DOS commands, from a virtual drive D, which is your user directory. To login type :

Service: < username >
userid: < username >
password: < password >

As you copy the files that you need onto your PC or onto your diskettes you can change the names of the robot programs at the same time. You can only send files into the Robot Controller that begin with HP (HoofdProgramma) followed by two digits (for example HP01 or HP99, which are the lower and upper limits).

Remark !!
Not all numbers between 0 and 99 are available for you. Ask (the robot manager) which number you can use.

Perhaps an other communications program is operational (when you must download a program). Then this paragraph is not correct.

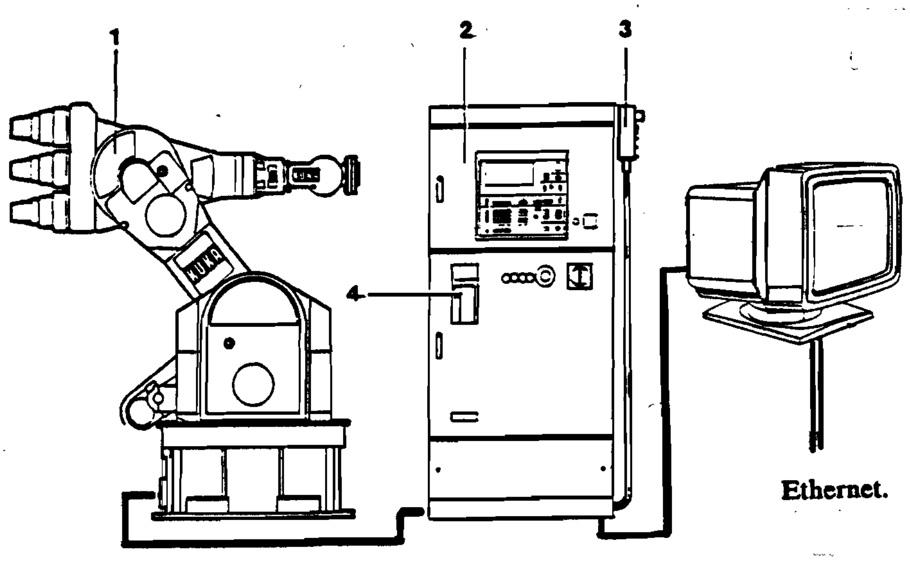


Figure 3.5. Robot system IR 161/15/25 and PC.

3.8. Downloading the Options file.

It is essential that the PC connected to the Robot Controller, has its own copy of the OPT-file, which is used to make the robot programs. This is necessary to correctly associate each timing measurement with the corresponding robot movement. It is advisable to place the OPT-file in the same directory on the PC, where the CTA PC-module is placed.

Downloading the OPT-file is similar to downloading the robot programs.

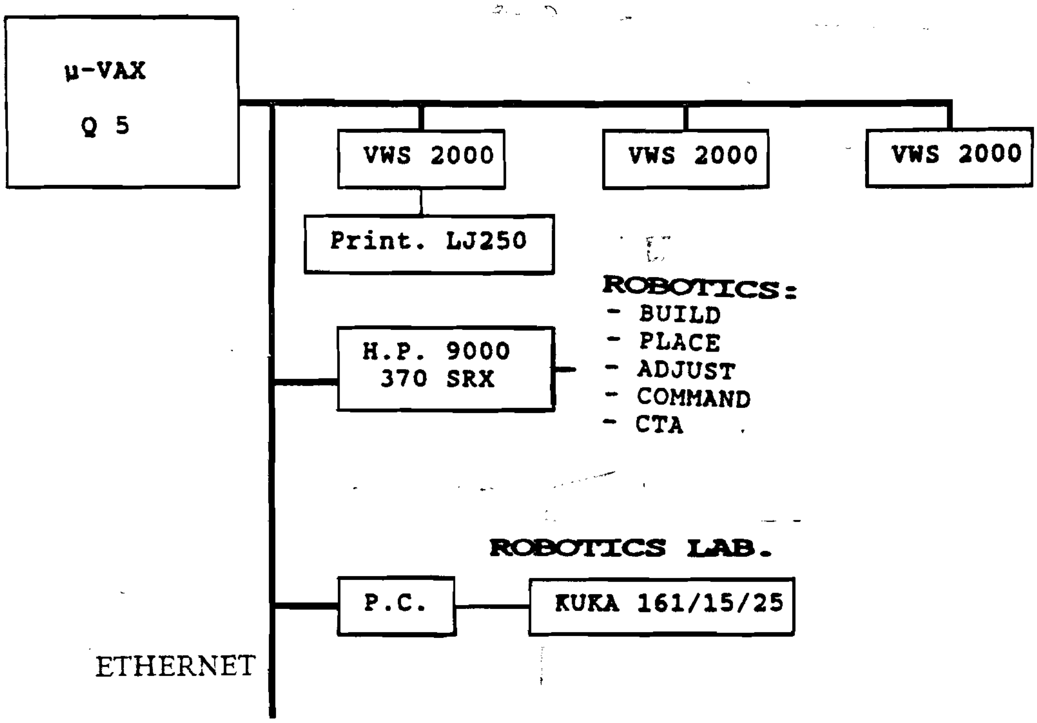


Figure 3.6. Computer structure WPA-CAD center.

3.9. Sending SRC-files to the Robot Controller.

The sending of files from the PC to the Robot Controller is done with DNC. DNC stands for Direct Numerical Control and is a product of KUKA. DNC is developed for off-line programming.

DNC starts up by typing DNC. Probably it does not matter where you are on the PC, otherwise you have to start it from C:. You must enter where the files, which DNC must send, are. Before you can send a file, the DNC communication line must be "open". Whether the DNC is "ON" or "OFF" is in the upper left corner of your screen. The Robot Controller must be in "EXTERN BEDRIJF". (For more information see Robot manual). Extern Bedrijf is under the A key on the control panel.

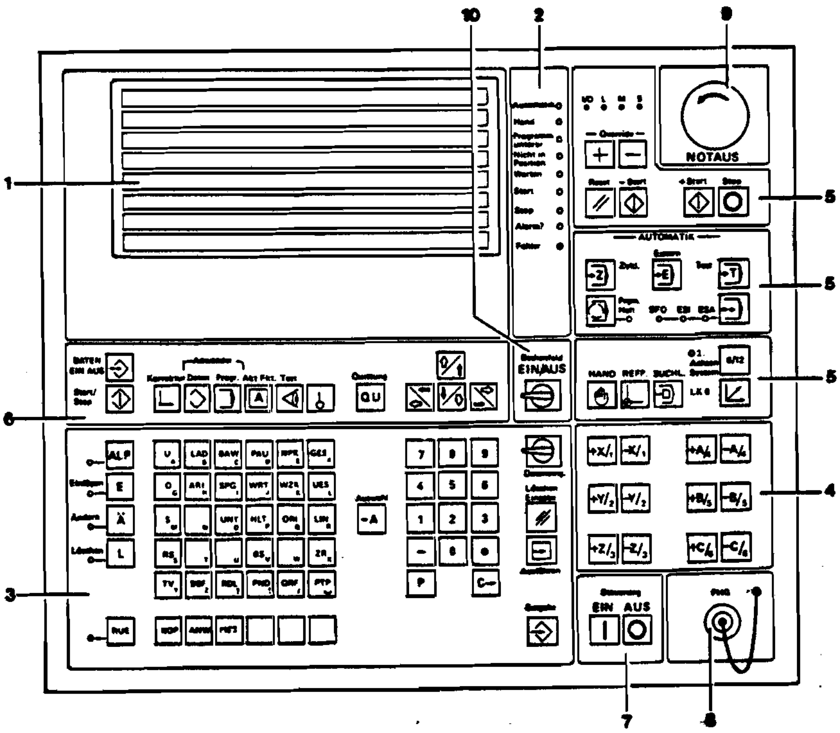


Figure 3.7. Control panel.

When "EXTERN BEDRIJF" is found, using the arrow keys, the connection is still not open. You have to press the key V for change (veranderen). Now press the "<-K" key to change the IN/UIT and press the "INGAVE" key to accept. You will hear a beep from the PC for changing from "DNC OFF" to "DNC ON". Use the menu options on the PC screen to send. To stop the "EXTERN BEDRIJF" on the Robot Controller use the key "STOP" under the emergency stop. Now the program is loaded and the Controller is free again.

A serious error occurred during the sending of the first SRC-files. The translation Eprom, which translates the ASCII robot commands into machine commands, was an old version and had a translation error. The "Z U #" command did not work. This Eprom was replaced by a new and better one. Then another error occurred. The $WISTAT (wrist status) statement of ROBOTICS was not identical to the $WISTAT statement of the Robot Controller.

With a PTP or a LIN statement the robot reaches a point in space. However the configuration of the robot arm is not determined when only a cartesian position is given. There are several arm configurations possible reaching one certain point in space. To reach a certain point in space with a certain arm configuration requires a statement containing the arm position together with the cartesian position of the point. This statement is the $WISTAT command. The $WISTAT command determines the robot work area and the robot arm configuration.

$WISTAT_T(BAA 1P 2N 3P 4N 5N 6P)
is such a WISTAT command. BAA determines the robot work area. BAA stands for Basic work Area and OVA stands for Overhead work Area. 1P to 6P determines the robot arm configuration. The total range of motion of a joint (axis constraints) is divided into a positive and a negative area with regard to the middle of the range. The P stands for the positive side of the middle and the N stands for the negative side of the middle.

There are three ways to define the WISTAT statement (they are all the same) :
- $WISTAT_T(OVA 1N 2N 3P 4N 5N 6P) is the text version.
- $WISTAT_D(91) is the decimal version.
- $WISTAT_H(5A) is the hexadecimal version.

It is advisable not to use the Text version of the WISTAT statement. This can cause translation errors when translating it to a KUKA robot file. It is better to use the WISTAT_D or the WISTAT_H statements, because numbers are always translated properly. The decimal and the hexadecimal wrist status are determined as follows : P = 0
N = 1
BAA = 0
OVA = 1.

The parameters are transformed into 0's and 1's. The first axis (1N) will be the least significant bit (first bit), the second axis (2N) the second bit (2^1), the third axis (3P) the third bit (2^2), etc.. The work area parameters will be the 64 bit (2^6).

Decimal :

OVA	1N	2N	3P	4N	5N	6P		
1	0	1	1	0	1	1		
2^6	2^5	2^4	2^3	2^2	2^1	2^0		
64	32	16	8	4	2	1		
64	0	16	8	0	2	1	=	91

Hexadecimal :

	OVA	1N	2N	3P	4N	5N	6P		
	1	0	1	1	0	1	1		
	2^2	2^1	2^0	2^3	2^2	2^1	2^0		
	4	2	1	8	4	2	1		
0	4	0	1	8	0	2	1		
	5			A				=	5A

When I entered the dutch robot programs (SRC-files) in the Robot Controller and started these programs, the robot reached the right cartesian positions but with the wrong arm configurations. I changed the arm configurations manually on the Robot Controller control panel, so that the robot moves to the positions with the right arm configuration. When comparing the two programs, the postprocessed and translated ROBOTICS program and the manually changed working robot program, which performs the same movement, I found that only the WISTAT statements were not the same. So probably ROBOTICS has an error in defining the wrist status. This error is maybe in the BUILD (BLD) file.

For the seven robot programs, I changed the $WISTAT-commands. To avoid a long list of robot programs, only the changed commands are summed up below :

```
HP91
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)  --> $WISTAT_D(0)
HP92
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)  --> $WISTAT_D(24)
$WISTAT_T(BAA 1N 2N 3N 4P 5N 6P)  --> $WISTAT_D(114)
HP93
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)  --> $WISTAT_D(116)
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)  --> $WISTAT_D(48)
HP94
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)  --> $WISTAT_D(0)
HP95
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)  --> $WISTAT_D(56)
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)  --> $WISTAT_D(40)
HP96
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6N)  --> $WISTAT_D(56)
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)  --> $WISTAT_D(24)
HP97
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)  --> $WISTAT_D(24)
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)  --> $WISTAT_D(56)
```

Remark !
Always check if the program, that you just entered into the Robot Controller, does exactly what you want it to do. If you run it immediately, serious accidents can happen.

3.10. Collecting the timing data.

In order to use the PC to time the robot moves, it is necessary to connect a Robot Controller output (Controller output port number 30, see Appendix 11.) via the ROBOT/PC-interface (see Appendix 12.) to the communications port on the PC. The interface changes the Robot Controller I/O signal to an interrupt that can be received by the PC (for schematic of the interface see appendix 11.).

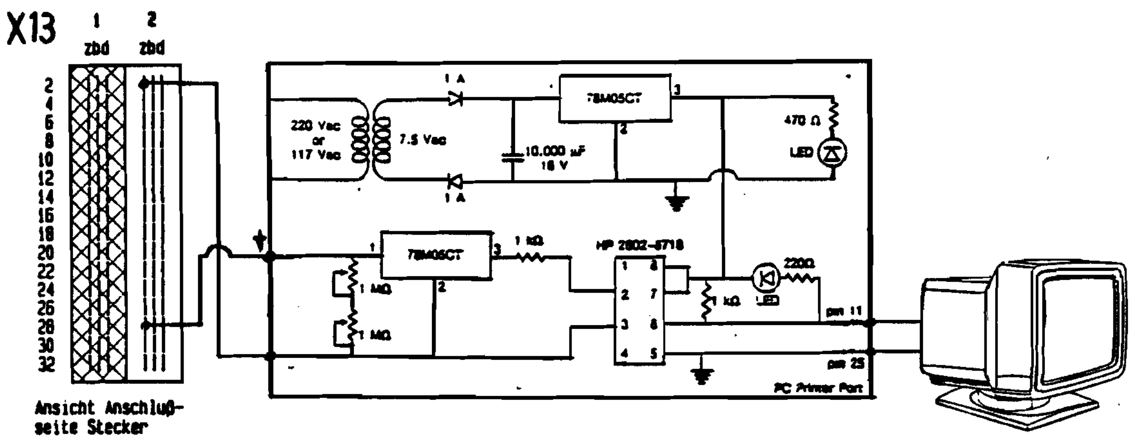


Figure 3.8. Robot Controller/PC connection.

Once the robot is setup and ready to run, the CTA-module which is on PC must be executed. Make sure you are in a directory which contains the OPT-file and has room for the separate timing data files that CTA creates. Type :

CTA

Now you are prompted to enter the name of the OPT-file. If CTA finds it, the main menu is displayed.

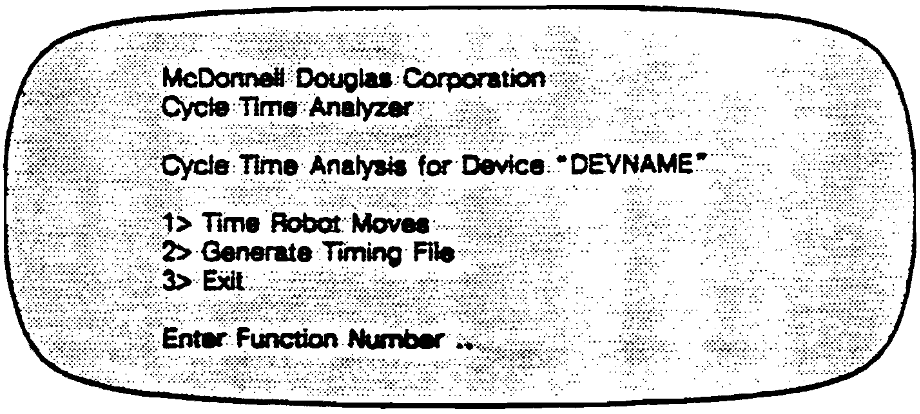


Figure 3.9. Cycle Time Analyzer main menu.

To begin the timing process, select option 1. on the menu. If the FOR LOOP parameter in the OPT-file is TRUE, the following menu is then displayed.

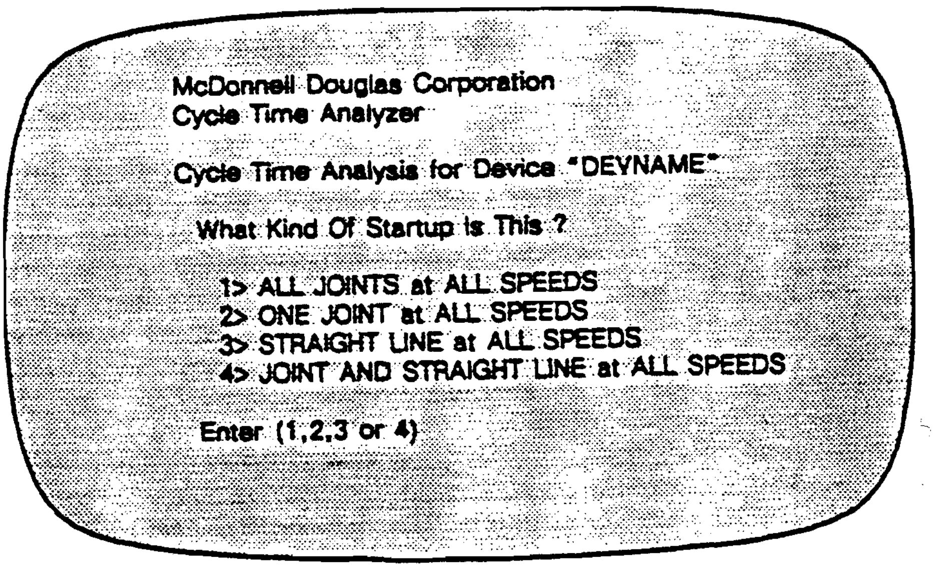


Figure 3.10. CTA startup menu (FOR LOOP = TRUE).

If the FOR LOOP parameter in the OPT-file is FALSE, the following menu is then displayed.

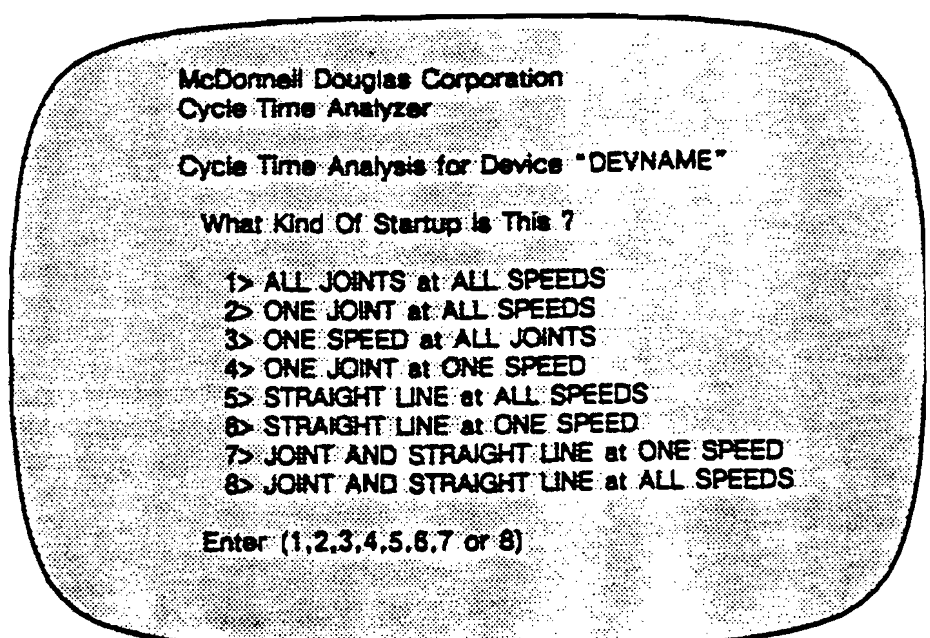


Figure 3.11. CTA startup menu (FOR LOOP = FALSE).

In order to create the TIM-file (see Appendix 13.), the timing data must be collected for all joints (including the straight line) at all speeds. The above menus show that there are several ways to organize the timing data collection. Select one of the valid menu items. Press any other key to go back to the main menu. Depending on which menu item you select, CTA will either instruct you which program to run on your robot, or ask you to enter the joint (and/or speed) which you want to time.

You should always visually check at the beginning of a timing session to make sure the PC is in fact timing the robot move. This is easily done by watching the light on the ROBOT/PC interface box and make sure it comes on when the robot starts a move and shuts off when it is finished.

The PC will display the duration of the movements as they are determined. CTA stores the timing data in a file with a DAT extension.

Once all the timing data has been collected for each joint and straight line set at all speeds, the TIM-file can be generated. Select on the main menu item 2 :

GENERATE TIMING FILE.

After this input, CTA processes all timing data, which are stored in the DAT-files, and creates the TIM-file. CTA displays informational messages while processing the timing data.

Remarks!

It is recommended that you select ONE JOINT AT ALL SPEEDS, so each joint is tested and then the comparable straight line options. Doing each one of these processes separately helps you to keep things a little more organized.

The timing of all the movements, at all speeds and for all joints, will take about four hours.

3.11. The timing file.

The timing file (TIM-file, see Appendix 13.) is set up to contain timing parameters for each joint of a robot over several speed settings. The TIM-file for the KUKA contains timing data for ten speed settings, from 10% to 100%, at 10% intervals. Therefore, for the six axes KUKA, there are sixty sets of data for joint interpolated motion. Also included is data for straight line motion for each of the ten speed settings.

The TIM-file is formatted as follows:

Line 1. contains seven pieces of information to joint motion:

<Min Jt Spd> <Max Jt Spd> <Num Jt> <Num Spd> <Spd Factor> <Units> <Jt CRD File>.

Where:

<Min Jt Spd> =
MINIMUM JOINT SPEED- the smallest acceptable speed stetting for joint interpolated motion.
<Max Jt Spd> =
MAXIMUM JOINT SPEED- the largest acceptable speed setting for joint interpolated motion.
<Num Jt> =
NUMBER OF JOINTS- the number of joints of the robot. There must be speed and acceleration data for each joint.
<Num Spd> =
NUMBER OF SPEEDS- the number of speeds for which there is timing data.
<Spd Factor> =
SPEED FACTOR- the interval size between each speed setting.
<Units> =
UNITS- a string defining the units of the joint interpolated speed setting.
<Jt CRD File> =
COORDINATE SYSTEM- the name of a CRD-file, which defines the relationship between the PLACE joint angles and the angles to which the speeds and accelerations apply.

Line 2. contains five pieces of information related to straight line motion:

<Min St Spd> <Max St Spd> <Spd Factor> <Units> <St CRD File>

Where :

<Min St Spd> =
MINIMUM STRAIGHT SPEED- the smallest acceptable speed setting for straight line motion.
<Max St Spd> =
MAXIMUM STRAIGHT SPEED- the largest acceptable speed setting for straight line motion.
<Spd Factor> =
SPEED FACTOR- the interval size between each speed setting.
<Units> =
UNITS- a string defining the units of the straight line speed setting.
<St CRD File> =
COORDINATE SYSTEM- the name of a CRD-file, which defines the cartesian position of the robot.

The rest of the file contains the timing data :

<Dwl Time> <Short Acc> <Long Acc> <Max Vel>

Where :

<Dwl Time> =
DWELL TIME- the length of time used before the move begins.
<Short Acc> =
SHORT ACCELERATION- the acceleration for short moves.
<Long Acc> =
LONG ACCELERATION- the acceleration for long moves.
<Max Vel> =
MAXIMUM VELOCITY- the maximum velocity.

This is the actual timing data used to calculate cycle times. There must be one line of data for each joint at each speed plus one set of straight line data for each speed. The units of the accelerations and velocities are determined by the CRD-files being used.

The timing data is ordered in such a way that the data for joint 1 (all speeds) is placed first. This is followed by joint 2 (all speeds). After the data for the last joint is the straight line data (all speeds):

<Min Jt Spd> <Max Jt Spd> <Num Jt> <Num Spd> <Spd Factor> <Units>
<Jt CRD File>.
<Min St Spd> <Max St Spd> <Spd Factor> <Units> <St CRD File>

<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Jt 1, Speed 1.
<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Jt 1, Speed 2.

<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Jt 1, Speed m.
<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Jt 2, Speed 1.

<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Jt n, Speed m.
<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Strt, Speed 1.

<Dwl Time> <Short Acc> <Long Acc> <Max Vel> Strt, Speed m.

3.12. Transferring the timing file back to the workstation.

After the TIM-file has been generated on the PC, it must be transferred to the workstation. This is the same procedure as downloading a robot program but then in the other way around. From the PC, via STARTNET, to your VAX user directory. Then copy the TIM-file to the PLACE system library, in order for all PLACE-users to access it. In addition, be sure that the necessary CRD-files are also in the PLACE system library.

Each time, during a PLACE-session, a device is merged into a cell, PLACE searches for a TIM-file with the same name as the robot's DCI-file. If a TIM-file is found, its cycle time model is used whenever that device is moved. Whenever a TIM-file is being used the symbol " ↑ " appears after the device name in the joints display window.

Chapter 4. CONCLUSIONS AND RECOMMENDATIONS.

Running CTA in it self was and is not a big problem. Because this was the first time CTA has been performed, many, often small, problems occurred.

Some problems have occurred trying to run the options file :
- There was no information about the use of lower and uppercase characters. Using lower case characters caused empty sequences.
- Writing an OPT-file from scratch instead of editing an existing one. Writing an options file from scratch in an editor caused non traceable errors.
- There was very little information about the syntax of the data for the straight line movement. The values of the position of the robot for straight motion are in the "movetext window".

Some problems occurred writing an user file.
- There was little information in the CTA-manual about the use of % or & in front of the lines.
- It was hard to find out what the syntax of the USR-file should be.
- Little information on functions which had to be used.

Some problems occurred trying to send SRC-files into the Robot Controller.
- A translation error in the postprocessor (JMP instead of SPG).
- There were differences in syntax on the Robot Controller and the robot program manual, due to an old Eprom in the Robot Controller.
- Some commands were not executable because of hardware errors on the Robot Controller translation Eprom.

During the testing of the actual robot programs, which were entered in the Robot Controller, a problem occurred : the $WISTAT command (wrist status) of ROBOTICS was not equal to the $WISTAT command of the Robot Controller. All the programs had to be checked in the Robot Controller and all $WISTAT commands had to be edited in an editor or at the robot.

Recommendations:

The documentation of ROBOTICS and the FALC need a lot of attention.

Test a few cycle times on the robot and in ROBOTICS. Program a certain movement in ROBOTICS and predict the cycle time. Send the program to the robot and measure the real cycle time. Do this for different loading conditions.

Make a small guide for the different steps in making an off-line robot program, containing also technical information, like :
- DNC sending mode information (9600,E,8,2) and how to change it
- STARTNET, ETHERNET explanations
- HP/VAX connections.

A SRCL postprocessor which operates under UNIX is needed to postprocess the programs on the HP workstation. Find out whether this postprocessor is available or has to be written.

Try to program other robot types with ROBOTICS. Use the PLACE system library, where a lot of robot are available. On the TUE are a few other robot's, mostly ASEA's. Make a program for such a robot. When you want to program one of the ASEA's you will need the ABB OLP compiler. This compiler is the last step in sending the robot program to the robot. The compiler is not yet available on the TUE and has to be ordered first.

Connect the PC of the Robot Controller to Ethernet.

Try to actually weld a product off-line. Use, for example, the existing DAF products, which are transported to the robot by the transport system. Model one or a few products in UNIGRAPHICS and place them in PLACE. Make a sequence that welds a product an send it to the robot. You can increase the difficulty by changing the position of the product by turning the manipulator.

LITERATURE.

[1] BUILD user guide., McDonnell Douglas Corporation, Release 7.0, 1991.

[2] Cycle Time Analyzer user guide., McDonnell Douglas Corporation, Release 6.0, 1990.

[3] Cycle Time Analyzer user guide., McDonnell Douglas Corporation, Release 7.0, 1991.

[4] COMMAND user guide. McDonnell Douglas Corporation, Release 7.0, 1991.

[5] An evaluation of the McDonnell Douglas Robotics 7.0 software., H.J. Van Veldhoven, Eindhoven, 7 June 1991. WPA number 1089.

The ROBOTICS Cycle Time Analyzer.
The first "time".

Appendices of WPA Nr : 1193.
M.C. Willems.

In order of : TUE-WPA

Professor : Prof. Dr. Ir. A.C.H. van der Wolf

Coaches : Ing. J.J.M. Schrauwen
: F. Soers

Author : M.C. Willems

Eindhoven, 8 november 1991.

SUMMARY.

Off-line programming of production machines is becoming increasingly important nowadays. Many software programs are developed for off-line programming. It is important that the software models of the production machines, which you are modeling, imitate the real production machines very accurately. ROBOTICS is such an off-line program package and this program has a module, Cycle Time Analyzer, for the dynamic calibration of a robot.

With CTA the total work area, for every axis, for the whole speed range of a robot is examined and stored in a file. The file is then connected to the robot in the software package, to predict accurate cycle times during a simulation.

The following actions are necessary to run the Cycle Time Analyzer.

- Write an options file. This file is the basis of your test. It contains the initial positions of the robot for every axes and the number and lengths of the test moves. It also contains the initial position of the robot, the number and length of the straight line movement.

- Run CTA on the HP workstation. CTA creates a cell and seven sequences: six sequences for the six different axes and one for the straight line move. The cell contains the robot (= a device) and the tpoints for the straight line move. These sequences perform the moves which are tested.

- Write an USR-file. The USR-file is the skeleton of your robot program. It contains the commands to turn a signal line "ON" and "OFF". This signal is used to determine the time of a test move. It also contains the commands of a loop to automatically cycle through the different robot speeds, during the tests.

- Run COMMAND on the HP workstation. In COMMAND, the USR-files and the sequences are processed into CSP-files.

- Run COMMAND on the VAX. In COMMAND on the VAX, the CSP-files are postprocessed into robot programs (SRC-files). These are in german. Translate them to dutch in an editor on the VAX. Change, if necessary, the $WISTAT commands.

- Download the SRC-files. Download the dutch SRC-files to the Robot Controller, via a communications program and ethernet. Use DNC to send them into the Robot Controller. The names of the programs which are send to the Robot Controller must exist of the characters HP and a two digit number.

- Run CTA on the PC. The CTA-PC module will cycle you through the determination of the timing data of all axes and the straight line movement and will produce the timing file (TIM-file).

- Transfer the TIM-file back to the workstation. Place the TIM file is the system library, in order for all users to access it.

Running CTA in it self was and is not a big problem. But many small problems had to be sorted out before the actual CTA was performed. This was due to the fact that it was the first "time".

CONTENTS

Appendix 1.

Appendix 1. 1. The KUKA BLD file.

The file, which was made of the KUKA-robot in the BUILD module.

;****** BUILD Release 7.0 ******

DEVICE NAME = KUKA
DEVICE TYPE = ROBOT
UNITS = MILLIMETERS

****

Constant
Translation along
Z axis
Amount = 374.6500 (MM)

****

Variable
Rotation about
Z axis
Joint Name = AXIS1
Joint Constraints --
High Value = 160.0000 (DEG) Low Value = -160.0000 (DEG)
Home Position = 0.0000 (DEG)
Joint Speed = 136.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK

****

Constant
Translation along
Z axis
Amount = 400.3040 (MM)

****

Constant
Rotation about
Y axis
Amount = -40.0000 (DEG)

****

Variable
Rotation about
Y axis
Joint Name = AXIS2
Joint Constraints --
High Value = 110.0000 (DEG) Low Value = -19.0000 (DEG)
Home Position = 0.0000 (DEG)

Joint Speed = 97.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK

****

Constant
Translation along
Z axis
Amount = 800.1000 (MM)

****

Constant
Rotation about
Y axis
Amount = 40.0000 (DEG)

****

Variable
Rotation about
Y axis
Joint Name = AXIS3
Joint Constraints --
High Value = 5.0000 (DEG) Low Value = -265.0000 (DEG)
Home Position = 0.0000 (DEG)
Joint Speed = 148.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK

****

Constant
Translation along
X axis
Amount = 617.7280 (MM)

****

Variable
Rotation about
X axis
Joint Name = AXIS4
Joint Constraints --
High Value = 250.0000 (DEG) Low Value = -250.0000 (DEG)
Home Position = 0.0000 (DEG)
Joint Speed = 187.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK

****

Constant

Translation along
X axis
Amount = 182.3720 (MM)

****

Variable
Rotation about
Y axis
Joint Name = AXIS5
Joint Constraints --
High Value = 135.0000 (DEG) Low Value = -135.0000 (DEG)
Home Position = 0.0000 (DEG)
Joint Speed = 182.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK

****

Constant
Translation along
X axis
Amount = 141.6800 (MM)

****

Variable
Rotation about
X axis
Joint Name = AXIS6
Joint Constraints --
High Value = 270.0000 (DEG) Low Value = -270.0000 (DEG)
Home Position = 0.0000 (DEG)
Joint Speed = 225.0000 (DEG/SEC)
Joint Acceleration = 0.0000 (DEG/SEC/SEC)
END OF LINK
END OF DEVICE

****

INVERSE KINEMATICS DATA --
SOURCE -- STANDARD

****

CONFIGURATIONS --
REACH FORWARD =
REACH BEHIND = NOT VALID CONFIG
ELBOW ABOVE =
ELBOW BELOW = NOT VALID CONFIG
JT 5 NEGATIVE = JT 5 NEGATIVE
JT 5 POSITIVE = JT 5 POSITIVE
Automatic wrist configuration
Initial Configuration = 1

****

MOTION TYPES --
NUMBER OF TYPES = 3
STRAIGHT
JOINT
SLEW
HOME MOTION TYPE = JOINT

****

TOOL COORDINATE SYSTEM = KUKATOOL
MAX TOOL SPEED = 0.0000 (MM/SEC)
MAX TOOL ACCEL = 0.0000 (MM/SEC/SEC)

****

COORDINATE SYSTEM REPRESENTATIONS --
NUMBER OF COORDINATE SYSTEMS = 2
KUKACART = CARTESIAN
JOINTS = JOINTS

****

World to Robot Base Transformation --
Translations --
0.0000 0.0000 0.0000 (MM)
Rotations --
0.0000 0.0000 0.0000 (DEG)

****

Link Names --
Number of Links = 7
1. KUKA00
2. KUKA01
3. KUKA02
4. KUKA03
5. KUKA04
6. KUKA05
7. KUKA06

****

Part Names --
Number of Parts = 7
1. KUKA00
2. KUKA01
3. KUKA02
4. KUKA03
5. KUKA04
6. KUKA05
7. KUKA06

2. The KUKA DCI-file. The file containing the kinematics and the axis constraints of the KUKA-robot.

```
BEGIN/HEADER
;
;
ROBOT
LINKS = 7
END/HEADER
BEGIN/KINEMATICS
  6.00000
   1.00000    1.00000    1.00000
   1.00000    1.00000    1.00000
  1.00000    0.00000    0.00000    0.00000
  0.00000    1.00000    0.00000    0.00000
  0.00000    0.00000    1.00000  -30.51000
  0.00000    1.00000    3.00000
  0.00000    0.00000    0.00000
  1.00000    2.00000
 31.50000   31.50000    0.00000
  0.00000    0.00000    0.00000
  0.00000   90.00000  -90.00000
  0.00000    0.00000    0.00000
  1.00000    0.00000    0.00000   -5.57795
  0.00000    1.00000    0.00000    0.00000
  0.00000    0.00000    1.00000    0.00000
   0.00000 -130.00000  130.00000
   0.00000    0.00000    0.00000
  0.00000    1.00000    2.00000    3.00000    4.00000    5.00000    6.00000
 AXIS1    AXIS2    AXIS3    AXIS4    AXIS5    AXIS6
 14.00000
   1.00000    3.00000    6.00000    9.00000   11.00000   13.00000
   2.0000     3.0000    14.7500
   1.0000     3.0000  -999.9900
   2.0000     3.0000    15.7600
   1.0000     2.0000   -40.0000
   1.0000     2.0000  -999.9900
   2.0000     3.0000    31.5000
   1.0000     2.0000    40.0000
   1.0000     2.0000  -999.9900
   2.0000     1.0000    24.3200
   1.0000     1.0000  -999.9900
   2.0000     1.0000     7.1800
   1.0000     2.0000  -999.9900
   2.0000     1.0000     5.5780
   1.0000     1.0000  -999.9900
END/KINEMATICS
BEGIN/JOINT_CONSTRAINTS
 6
      1.0000     0.0000   160.0000    0.0000
      2.0000    45.5000    64.5000    0.0000
      3.0000  -130.0000   135.0000    0.0000
      4.0000     0.0000   250.0000    0.0000
      5.0000     0.0000   135.0000    0.0000
```

```
    6.0000     0.0000  270.0000     0.0000
END/JOINT_CONSTRAINTS
BEGIN/HOME
  0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
END/HOME
BEGIN/CONFIGURATION
FIXED FIXED AUTO
 1
 1. JT 5 NEGATIVE
 2. JT 5 POSITIVE
END/CONFIGURATION
BEGIN/JOINT_SPEED
  2.37365   1.69297   2.58309
  3.26377   3.17650   3.92699
END/JOINT_SPEED
BEGIN/JOINT_ACCEL
  0.00000   0.00000   0.00000
  0.00000   0.00000   0.00000
END/JOINT_ACCEL
BEGIN/TRAJECTORY
 2
 1
 2
 3
END/TRAJECTORY
BEGIN/TOOL_TIP_DEF
KUKATOOL
END/TOOL_TIP_DEF
BEGIN/MAX_TOOL_SPD
  0.00000
END/MAX_TOOL_SPD
BEGIN/TOOL_ACCEL
  0.00000
END/TOOL_ACCEL
BEGIN/CRD_SYS_REP
KUKACART CARTESIAN
JOINTS  JOINTS
END/CRD_SYS_REP
BEGIN/ADDITIONAL_KINEMATICS
 0.0
EXTKIN= NONE
CRD= NONE
JTMAPCRD= NONE
END/ADDITIONAL_KINEMATICS
```

3. The KUKA DEV-file. The file where the KUKA-robot is defined as a device.

```
;****** BUILD Release 7.0 ******
FRAMES
WORLD    WORLD
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA00   WORLD
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA01   KUKA00
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA02   KUKA01
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA03   KUKA02
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA04   KUKA03
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA05   KUKA04
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA06   KUKA05
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
END/FRAMES
CONTROL
KUKA     DEV KUKA     KUKA06
END/CONTROL
DISPLAY
KUKA00   KUKA00    WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA01   KUKA01    WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA02   KUKA02    WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA03   KUKA03    WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
```

```
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA04   KUKA04   WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA05   KUKA05   WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA06   KUKA06   WHITE,H(0.0),S(0.000),I(1)   TOLER(0.0500)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
END/DISPLAY
TPOINTS
KUKA06      WHITE,H(0.0),S(0.000),I(1)
1
TP1
   0.0000   0.0000   0.0000 1.0000 0.0000 0.0000 0.0000 1.0000 0.0000
END/TPOINTS
```

4. The TIJS CEL-file. The file containing the cell. This cell was especially made for CTA, containing only the KUKA-robot.

```
;****** PLACE Release 7.0 ******
FRAMES
WORLD    WORLD
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA00   WORLD
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA01   KUKA00
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  14.7500
KUKA02   KUKA01
   0.7660  0.0000  -0.6428  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.6428  0.0000  0.7660  15.7600
KUKA03   KUKA02
   0.7660  0.0000  0.6428  0.0000
   0.0000  1.0000  0.0000  0.0000
  -0.6428  0.0000  0.7660  31.5000
KUKA04   KUKA03
   1.0000  0.0000  0.0000  24.3200
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA05   KUKA04
   1.0000  0.0000  0.0000  7.1800
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA06   KUKA05
   1.0000  0.0000  0.0000  5.5780
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
TOORTS   KUKA06
   1.0000  0.0000  0.0000  0.0000
   0.0000  -0.7071  -0.7071  0.0000
   0.0000  0.7071  -0.7071  0.0000
END/FRAMES
CONTROL
KUKA     DEV KUKA     KUKA06
END/CONTROL
DISPLAY
KUKA00   KUKA00    ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA01   KUKA01    MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
```

```
   0.0000   0.0000   1.0000   0.0000
KUKA02    KUKA02     ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA03    KUKA03     MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA04    KUKA04     ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA05    KUKA05     MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA06    KUKA06     ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
TOORTS    TOORTS     MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
END/DISPLAY
TPOINTS
WORLD      WHITE,R(1.0000),G(1.0000),B(1.0000)
4
TPT1
   46.1024  -16.5354   25.9606 -0.7193  0.6947  0.0000  0.0000  0.0000 -1.0000
TPT2
   26.4480   14.0640   27.8080 -0.7678  0.6406  0.0013  0.0115  0.0158 -0.9998
TPT3
   60.2640   14.0640   27.0720 -0.7592  0.6509  0.0013  0.0117  0.0156 -0.9998
TPT4
   44.9008  -19.5081   60.2362 -0.9063  0.4226  0.0000  0.0000  0.0000 -1.0000
KUKA06     WHITE,R(1.0000),G(1.0000),B(1.0000)
1
TP1
   0.0000   0.0000   0.0000 1.0000  0.0000  0.0000  0.0000  1.0000  0.0000
TOORTS     WHITE,R(1.0000),G(1.0000),B(1.0000)
1
TP
   6.1024  10.6299   0.0000 1.0000  0.0000  0.0000  0.0000  1.0000  0.0000
END/TPOINTS
```

5. The JOINTSM CRD-file. The file containing the joint angles relationships.

```
COORD SYS NAME = JOINTSM;

COORD SYS TYPE = JOINT;

UNITS;
ROTATIONS = DEG;
TRANSLATIONS = MM;

NUMBER OF PARAMETERS = DOF;

ORDER : SAME AS JOINTS;

NAMES;
1 : J1;
2 : J2;
3 : J3;
4 : J4;
5 : J5;
6 : J6;
END NAMES;
```

6. The KUKACART CRD-file. The file for defining a cartesian position of the robot.

```
COORDINATE SYSTEM NAME = KUKACART;

COORDINATE SYSTEM TYPE = MATRIX;

UNITS;
ROTATIONS = DEG;
TRANSLATIONS = MM;

DEFINITION;
XYZ + ANGLES;
ANGLES = RZ,RY,RX;
TOOL = RY 90.0, RZ 180.0 ;

NUMBER OF PARAMETERS = 6;

ORDER;
1 = P1;
2 = P2;
3 = P3;
4 = P4;
5 = -1 * P5 + 180.0;
6 = -1 * P6 + 180.0;
END ORDER;

INVERSE;
1 = P1;
2 = P2;
3 = P3;
4 = P4;
5 = -1 * P5 + 180.0;
6 = -1 * P6 + 180.0;
END INVERSE;

LIMITS;
4 : HI = 180.0 , LO = -180.0 , UNITS = DEG ;
5 : HI =  90.0 , LO =  -90.0 , UNITS = DEG ;
6 : HI = 180.0 , LO = -180.0 , UNITS = DEG ;
END LIMITS;

NAMES;
1 : X;
2 : Y;
3 : Z;
4 : A;
5 : B;
6 : C;
END NAMES;
```

Appendix 2. The OPT-file.

```
DEVICE NAME = KUKA
DCI NAME = KUKA
JOINT CRD NAME = JOINTSM
STRAIGHT CRD NAME = KUKACART
OUTPUT NAME = TIJSJ
CELL OR DEVICE = CELL
CELL NAME = TIJS
NUMBER OF JOINTS = 6
NUMBER OF JOINT SPEEDS = 10
MIN JOINT SPEED = 0.00
MAX JOINT SPEED = 100.0
JOINT SPEED FACTOR = 10.0
JOINT SPEED UNITS = PCNT
MIN STRAIGHT SPEED = 0.0
MAX STRAIGHT SPEED = 100.0
STRAIGHT SPEED FACTOR = 10.0
CONVERSION FACTOR = 1.0
STRAIGHT SPEED UNITS = M/MIN
FOR LOOP = TRUE
DATA = -79.0  60.0  -56.0   10.0  10.0  125.0   0.0 20  1.0 14 10.0
DATA = -85.0  45.0  -40.0    0.0  -54.0  226.0   0.0 20  1.0 10 10.0
DATA = -85.0  40.0  -130.0   0.0  -53.0  226.0   0.0 20  1.0 22 10.0
DATA = -85.0  48.0  -50.0    0.0  10.0  134.0   0.0 20  1.0 32 10.0
DATA = -85.0  48.0  -50.0  180.0   2.0   44.0   0.0 20  1.0 20 10.0
DATA = -85.0  48.0  -50.0  180.0  10.0  -45.0   0.0 20  1.0 34 10.0
DATA = 320.1 -731.3 1216.2  36.4  -71.9   80.6   0.0 20 20.0 20 70.0
```

Appendix 3. The axis constraints.

The constraints of axis 1 are 160 degrees and -160 degrees, but because of the welding thread support post the range from 0 degrees to 160 degrees is not usable.

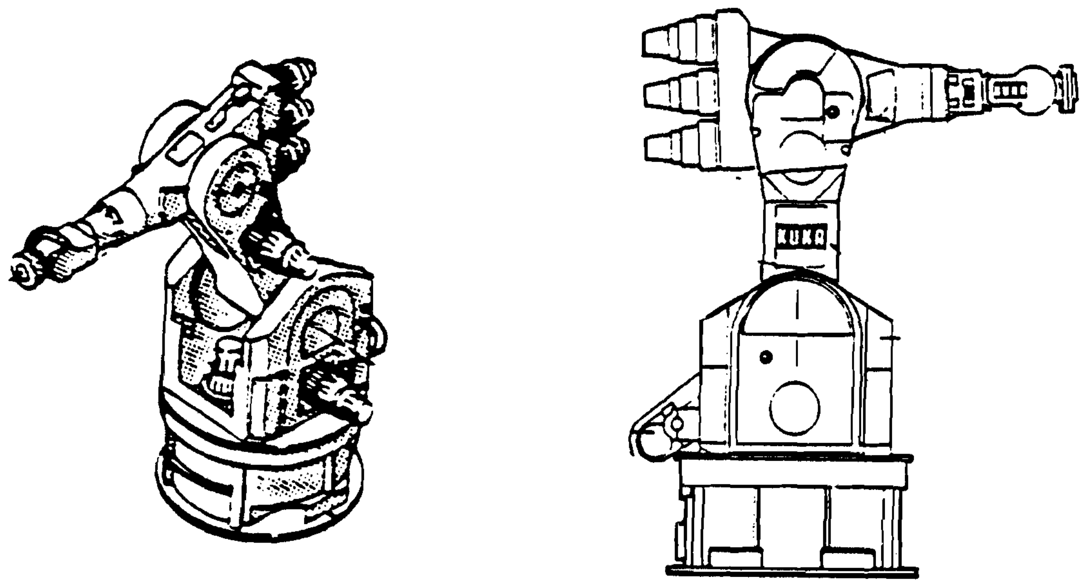


Figure A3.1. Axis 1. -159 degrees and 1 degree.

The constraints of axis 2 are 110 degrees and -19 degrees.

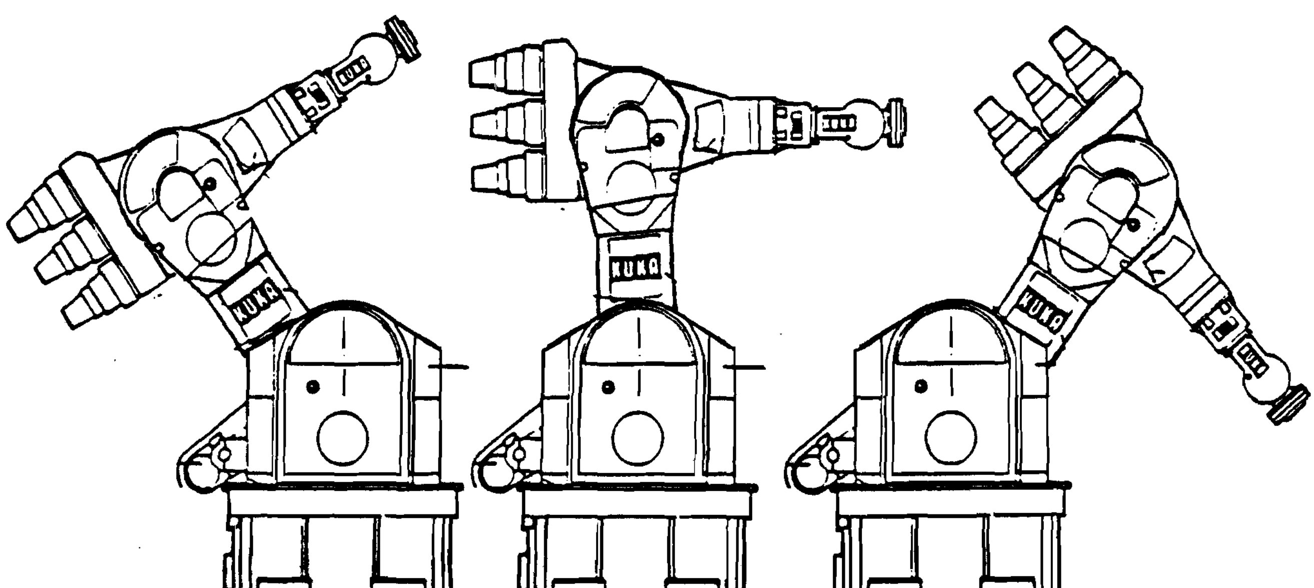


Figure A3.2. Axis 2. -15 degrees, 45 degrees and 105 degrees.

The constraints of axis 3 are 5 degrees and -265 degrees.

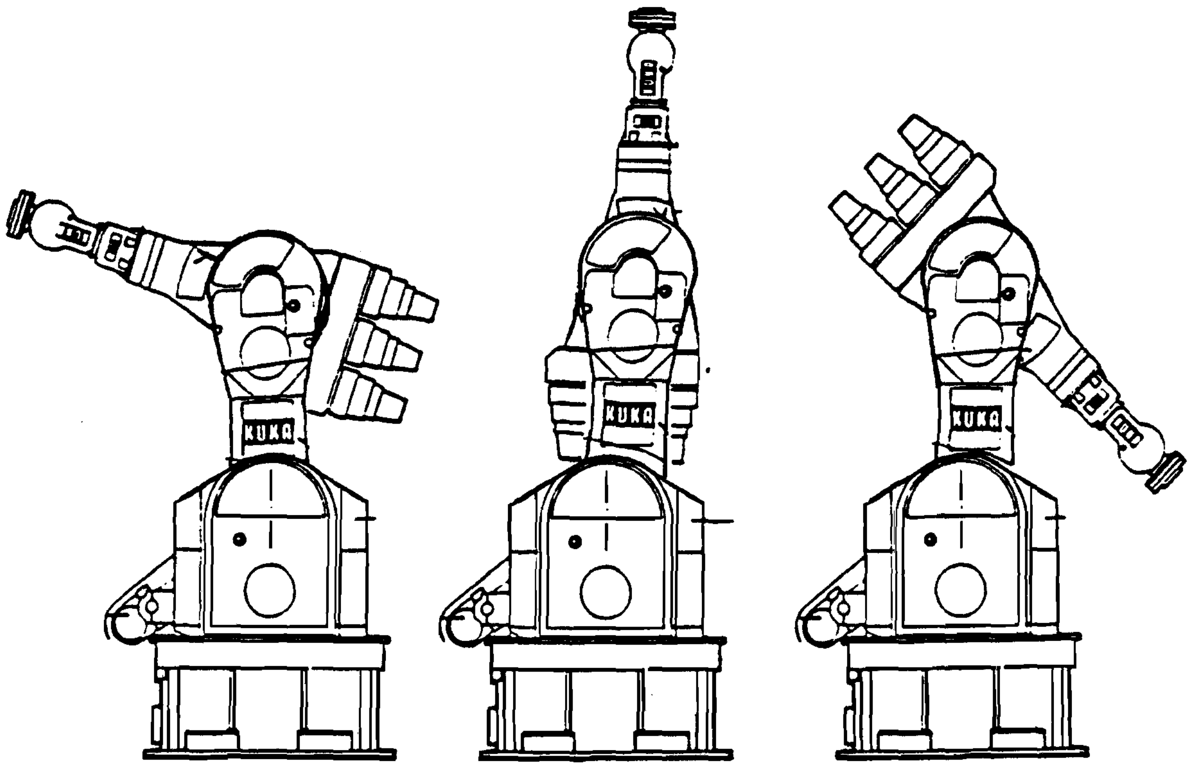


Figure A3.3. Axis 3. -10 degrees, -130 degrees and -250 degrees.

The constraints of axis 4 are 250 degrees and -250 degrees.

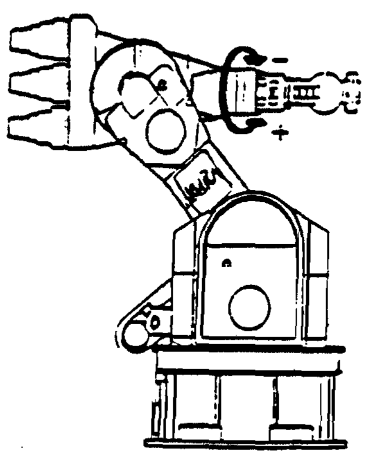


Figure A3.4. Axis 4. 170 degrees to -170 degrees.

The constraints of axis 5 are 135 degrees and -135 degrees.

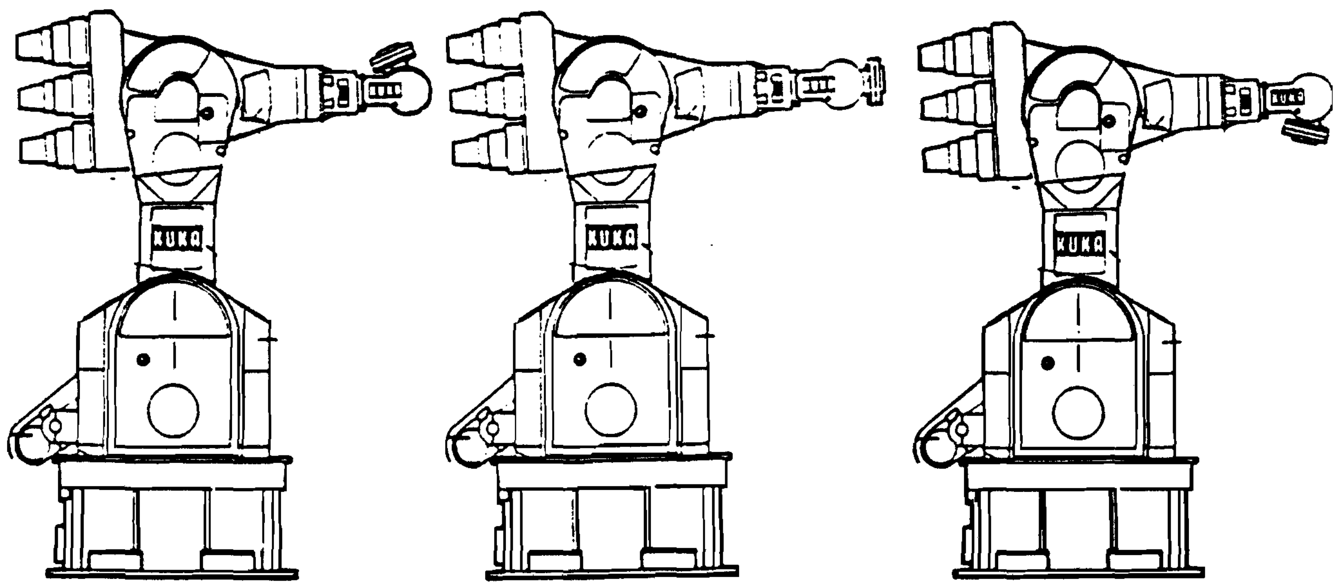


Figure A3.5. Axis 5. -108 degrees, 2 degrees and 112 degrees.

The constraints of axis 6 are 270 degrees and -270 degrees.

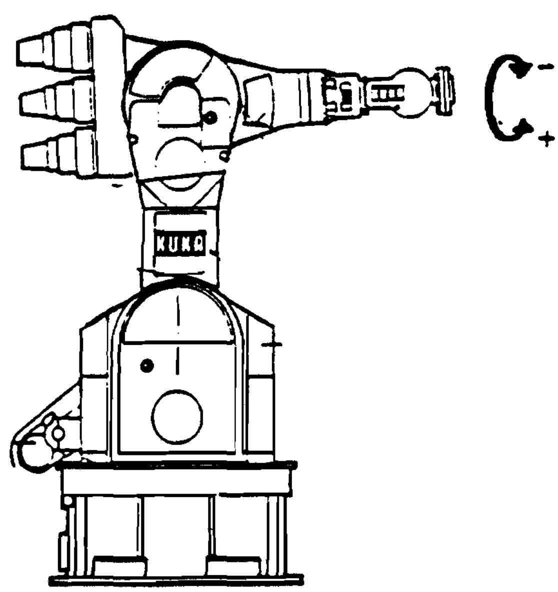


Figure A3.6. Axis 6. 135 degrees to -225 degrees.

Appendix 4. The CEL-file (TIJSS.CEL) generated by the CTA-module.

```
;****** PLACE Release 7.0 ******
FRAMES
WORLD     WORLD
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA00    WORLD
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA01    KUKA00
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  14.7500
KUKA02    KUKA01
   0.7660  0.0000  -0.6428  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.6428  0.0000  0.7660  15.7600
KUKA03    KUKA02
   0.7660  0.0000  0.6428  0.0000
   0.0000  1.0000  0.0000  0.0000
  -0.6428  0.0000  0.7660  31.5000
KUKA04    KUKA03
   1.0000  0.0000  0.0000  24.3200
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA05    KUKA04
   1.0000  0.0000  0.0000  7.1800
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA06    KUKA05
   1.0000  0.0000  0.0000  5.5780
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
TOORTS    KUKA06
   1.0000  0.0000  0.0000  0.0000
   0.0000  -0.7071  -0.7071  0.0000
   0.0000  0.7071  -0.7071  0.0000
END/FRAMES
CONTROL
KUKA    DEV KUKA    KUKA06
END/CONTROL
DISPLAY
KUKA00    KUKA00    ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
KUKA01    KUKA01    MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000  0.0000  0.0000  0.0000
   0.0000  1.0000  0.0000  0.0000
   0.0000  0.0000  1.0000  0.0000
```

```
KUKA02    KUKA02    ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA03    KUKA03    MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA04    KUKA04    ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA05    KUKA05    MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
KUKA06    KUKA06    ORANGE,R(1.0000),G(0.5294),B(0.0000) TOLER(0.0500) TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
TOORTS    TOORTS    MEDIUMFORESTGREEN,R(0.1961),G(0.5059),B(0.2941) TOLER(0.0500)
TRANSP(1.0000)
   1.0000   0.0000   0.0000   0.0000
   0.0000   1.0000   0.0000   0.0000
   0.0000   0.0000   1.0000   0.0000
END/DISPLAY
TPOINTS
WORLD     WHITE,R(1.0000),G(1.0000),B(1.0000)
4
TPT1
  46.1024 -16.5354  25.9606 -0.7193  0.6947  0.0000  0.0000  0.0000 -1.0000
TPT2
  26.4480  14.0640  27.8080 -0.7678  0.6406  0.0013  0.0115  0.0158 -0.9998
TPT3
  60.2640  14.0640  27.0720 -0.7592  0.6509  0.0013  0.0117  0.0156 -0.9998
TPT4
  44.9008 -19.5081  60.2362 -0.9063  0.4226  0.0000  0.0000  0.0000 -1.0000
KUKA00    WHITE,R(1.0000),G(1.0000),B(1.0000)
41
TPT
  12.6024 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT1
  13.3898 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT2
  11.8150 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT3
  14.1772 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT4
  11.0276 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT5
  14.9646 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
```

TPT6
10.2402 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065

TPT7
15.7520 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT8
9.4528 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT9
16.5394 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT10
8.6654 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT11
17.3268 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT12
7.8780 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT13
18.1142 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT14
7.0906 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT15
18.9016 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT16
6.3031 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT17
19.6890 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT18
5.5157 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT19
20.4764 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT20
4.7283 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT21
23.2323 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT22
1.9724 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT23
25.9882 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT24
-0.7835 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT25
28.7441 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT26
-3.5394 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT27
31.5000 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT28
-6.2953 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT29
34.2559 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT30
-9.0512 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT31
37.0118 -28.7913 47.8819 -0.7104 0.7020 -0.0507 0.6579 0.6879 0.3065
TPT32

```
 -11.8071 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT33
 39.7677 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT34
 -14.5630 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT35
 42.5236 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT36
 -17.3189 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT37
 45.2795 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT38
 -20.0748 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT39
 48.0354 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
TPT40
 -22.8307 -28.7913  47.8819 -0.7104  0.7020 -0.0507  0.6579  0.6879  0.3065
KUKA06     WHITE,R(1.0000),G(1.0000),B(1.0000)
2
TP1
  0.0000   0.0000   0.0000  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000
TPW
  0.0000   0.0000   0.0000  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000
TOORTS     WHITE,R(1.0000),G(1.0000),B(1.0000)
1
TP
  6.1024  10.6299   0.0000  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000
END/TPOINTS
```

Appendix 5. The SEQ-files generated by CTA.

Appendix 5. TIJSJ1.SEQ, the sequence for axis 1.

```
;****** PLACE Release 9.0 ******
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-79.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, NOP;
GOTO_JOINTS: (IN),-78.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-80.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-77.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-81.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-76.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-82.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-75.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-83.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-74.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-84.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-73.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-72.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-86.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-71.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-87.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-70.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-88.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-69.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-89.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-59.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-99.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-49.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-109.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-39.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-119.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-29.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-129.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-19.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-139.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-9.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-149.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),1.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-159.0000,60.0000,-56.0000,10.0000,10.0000,125.0000, OUTLAY;
```

Appendix 5. TIJSJ2.SEQ, the sequence for axis 2.

```
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-85.0000,45.0000,-40.0000,0.0000,-54.0000,226.0000, NOP;
GOTO_JOINTS: (IN),-85.0000,46.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,44.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,47.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,43.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,42.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,49.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,41.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,50.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,51.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,39.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,52.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,38.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,53.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,37.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,54.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,36.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,55.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,35.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,65.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,25.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,75.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,15.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,85.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,5.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,95.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,-5.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,105.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,-15.0000,-40.0000,0.0000,-54.0000,226.0000, OUTLAY;
```

Appendix 5. TIJSJ3.SEQ, the sequence for axis 3.

```
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-85.0000,40.0000,-130.0000,0.0000,-53.0000,226.0000, NOP;
GOTO_JOINTS: (IN),-85.0000,40.0000,-129.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-131.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-128.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-132.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-127.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-133.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-126.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-134.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-125.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-135.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-124.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-136.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-123.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-137.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-122.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-138.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-121.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-139.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-120.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-140.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-110.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-150.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-100.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-160.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-90.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-170.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-80.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-180.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-70.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-190.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-60.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-200.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-50.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-210.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-40.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-220.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-30.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-230.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-20.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-240.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-10.0000,0.0000,-53.0000,226.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,40.0000,-250.0000,0.0000,-53.0000,226.0000, OUTLAY;
```

Appendix 5. TIJSJ4.SEQ, the sequence for axis 4.

```
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,0.0000,10.0000,134.0000, NOP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,1.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-1.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,2.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-2.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,3.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-3.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,4.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-4.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,5.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-5.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,6.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-6.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,7.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-7.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,8.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-8.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,9.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-9.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,10.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-10.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,20.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-20.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,30.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-30.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,40.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-40.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,50.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-50.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,60.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-60.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,70.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-70.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,80.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-80.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,90.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-90.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,100.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-100.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,110.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-110.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,120.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-120.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,130.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-130.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,140.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-140.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,150.0000,10.0000,134.0000, OUTLAY;
```

```
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-150.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,160.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-160.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,170.0000,10.0000,134.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,-170.0000,10.0000,134.0000, OUTLAY;
```

Appendix 5. TIJSJ5.SEQ, the sequence for axis 5.

```
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,2.0000,44.0000, NOP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,3.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,1.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,4.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,0.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,5.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-1.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,6.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-2.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,7.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-3.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,8.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-4.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,9.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-5.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-6.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,11.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-7.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,12.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-8.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,22.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-18.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,32.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-28.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,42.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-38.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,52.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-48.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,62.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-58.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,72.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-68.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,82.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-78.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,92.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-88.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,102.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-98.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,112.0000,44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,-108.0000,44.0000, OUTLAY;
```

Appendix 5. TIJSJ6.SEQ, the sequence for axis 6.

```
;****** CTA Release 7.0 *****
BEGIN_SEGMENT: STARTUP;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-45.0000, NOP;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-44.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-46.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-43.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-47.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-42.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-48.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-41.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-49.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-40.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-50.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-39.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-51.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-38.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-52.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-37.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-53.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-36.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-54.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-35.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-55.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-25.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-65.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-15.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-75.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-5.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-85.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,5.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-95.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,15.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-105.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,25.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-115.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,35.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-125.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,45.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-135.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,55.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-145.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,65.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-155.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,75.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-165.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,85.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-175.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,95.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-185.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,105.0000, OUTLAY;
```

```
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-195.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,115.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-205.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,125.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-215.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,135.0000, OUTLAY;
GOTO_JOINTS: (IN),-85.0000,48.0000,-50.0000,180.0000,10.0000,-225.0000, OUTLAY;
```

Appendix 5. TIJSS.SEQ, the sequence for the straight line movement.

```
;****** PLACE Release 7.0 ******
BEGIN_SEGMENT: STARTUP;
WORKING_TPOINT: KUKA06,TPW;
SET_DEVICE_MOTION_MODE: INTERPOLATE;
END_SEGMENT: STARTUP;
GOTO_TPOINT: KUKA00,TPT,NOP;
SET_DEVICE_MOTION_MODE: STRAIGHT;
GOTO_TPOINT: KUKA00,TPT1,OUTLAY;
GOTO_TPOINT: KUKA00,TPT2,OUTLAY;
GOTO_TPOINT: KUKA00,TPT3,OUTLAY;
GOTO_TPOINT: KUKA00,TPT4,OUTLAY;
GOTO_TPOINT: KUKA00,TPT5,OUTLAY;
GOTO_TPOINT: KUKA00,TPT6,OUTLAY;
GOTO_TPOINT: KUKA00,TPT7,OUTLAY;
GOTO_TPOINT: KUKA00,TPT8,OUTLAY;
GOTO_TPOINT: KUKA00,TPT9,OUTLAY;
GOTO_TPOINT: KUKA00,TPT10,OUTLAY;
GOTO_TPOINT: KUKA00,TPT11,OUTLAY;
GOTO_TPOINT: KUKA00,TPT12,OUTLAY;
GOTO_TPOINT: KUKA00,TPT13,OUTLAY;
GOTO_TPOINT: KUKA00,TPT14,OUTLAY;
GOTO_TPOINT: KUKA00,TPT15,OUTLAY;
GOTO_TPOINT: KUKA00,TPT16,OUTLAY;
GOTO_TPOINT: KUKA00,TPT17,OUTLAY;
GOTO_TPOINT: KUKA00,TPT18,OUTLAY;
GOTO_TPOINT: KUKA00,TPT19,OUTLAY;
GOTO_TPOINT: KUKA00,TPT20,OUTLAY;
GOTO_TPOINT: KUKA00,TPT21,OUTLAY;
GOTO_TPOINT: KUKA00,TPT22,OUTLAY;
GOTO_TPOINT: KUKA00,TPT23,OUTLAY;
GOTO_TPOINT: KUKA00,TPT24,OUTLAY;
GOTO_TPOINT: KUKA00,TPT25,OUTLAY;
GOTO_TPOINT: KUKA00,TPT26,OUTLAY;
GOTO_TPOINT: KUKA00,TPT27,OUTLAY;
GOTO_TPOINT: KUKA00,TPT28,OUTLAY;
GOTO_TPOINT: KUKA00,TPT29,OUTLAY;
GOTO_TPOINT: KUKA00,TPT30,OUTLAY;
GOTO_TPOINT: KUKA00,TPT31,OUTLAY;
GOTO_TPOINT: KUKA00,TPT32,OUTLAY;
GOTO_TPOINT: KUKA00,TPT33,OUTLAY;
GOTO_TPOINT: KUKA00,TPT34,OUTLAY;
GOTO_TPOINT: KUKA00,TPT35,OUTLAY;
GOTO_TPOINT: KUKA00,TPT36,OUTLAY;
GOTO_TPOINT: KUKA00,TPT37,OUTLAY;
GOTO_TPOINT: KUKA00,TPT38,OUTLAY;
GOTO_TPOINT: KUKA00,TPT39,OUTLAY;
GOTO_TPOINT: KUKA00,TPT40,OUTLAY;
```

Appendix 6. The USR-files.

The USR-file for axis 1 to 6 are all the same, except for the program names and sequence names.

```
FUNCTION=ON,HP91
WISTAT=T
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
&OPERATION OUTLAY
S A 30
&INC_GOTO
RS A 30
WRT Z 5
&END_OPERATION
&REF_SEQ TIJSJ1
GES ALL P1
&INC_SEG STARTUP
&INC_GOTO
&INC_SEQ TIJSJ1
ADD P1 KON+10
WRT Z 100
JMP AD 5    (<- little error in the software)
```

Figure A6.1. The USR-file for axis 1.

```
FUNCTION=ON,HP97
WISTAT=T
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
&OPERATION OUTLAY
S A 30
&INC_GOTO
RS A 30
WRT Z 5
&END_OPERATION
&REF_SEQ TIJSJ7
GES BAN P1
&INC_SEG STARTUP
&INC_GOTO
&INC_SEQ TIJSJ7
ADD P1 KON+10
WRT Z 100
JMP AD 5    (<- little error in the software)
```

Figure A6.2. The USR-file for straight line movement.

Appendix 7. The CSP-files generated by COMMAND.

The CSP-files are very the same for all axis. Therefore, only the CSP-files for axis 1 and the straight line movement are included.

The KUKA CSP-file for axis 1.

```
&BEGIN/DEVICES;
& KUKA     6 KUKA;
&END/DEVICES;

&BEGIN/LOCATIONS KUKA;
&LOCATION: JV CARTESIAN
 234.9390 -1186.2666 1437.0870  -0.8036  43.6203  109.3018
EXTRA_LOC= -79.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV0 CARTESIAN
 255.6065 -1181.9857 1437.0870  0.1964  43.6203  109.3018
EXTRA_LOC= -78.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV1 CARTESIAN
 214.2001 -1190.1862 1437.0870  -1.8036  43.6203  109.3018
EXTRA_LOC= -80.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV2 CARTESIAN
 276.1960 -1177.3447 1437.0870  1.1964  43.6203  109.3018
EXTRA_LOC= -77.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV3 CARTESIAN
 193.3958 -1193.7432 1437.0870  -2.8036  43.6203  109.3018
EXTRA_LOC= -81.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV4 CARTESIAN
 296.7015 -1172.3451 1437.0870  2.1964  43.6203  109.3018
EXTRA_LOC= -76.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV5 CARTESIAN
 172.5327 -1196.9366 1437.0870  -3.8036  43.6203  109.3018
EXTRA_LOC= -82.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV6 CARTESIAN
 317.1165 -1166.9884 1437.0870  3.1964  43.6203  109.3018
EXTRA_LOC= -75.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV7 CARTESIAN
 151.6170 -1199.7655 1437.0870  -4.8036  43.6203  109.3018
EXTRA_LOC= -83.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV8 CARTESIAN
 337.4350 -1161.2762 1437.0870  4.1964  43.6203  109.3018
EXTRA_LOC= -74.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JV9 CARTESIAN
 130.6551 -1202.2288 1437.0870  -5.8036  43.6203  109.3018
EXTRA_LOC= -84.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JVA CARTESIAN
 357.6507 -1155.2103 1437.0870  5.1964  43.6203  109.3018
EXTRA_LOC= -73.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JVB CARTESIAN
 109.6534 -1204.3259 1437.0870  -6.8036  43.6203  109.3018
EXTRA_LOC= -85.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JVC CARTESIAN
 377.7574 -1148.7925 1437.0870  6.1964  43.6203  109.3018
EXTRA_LOC= -72.0000  60.0000  -56.0000  10.0000  10.0000  125.0000 ;
```

&LOCATION: JVD CARTESIAN
88.6183 -1206.0562 1437.0870 -7.8036 43.6203 109.3018
EXTRA_LOC= -86.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVE CARTESIAN
397.7490 -1142.0248 1437.0870 7.1964 43.6203 109.3018
EXTRA_LOC= -71.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVF CARTESIAN
67.5562 -1207.4192 1437.0870 -8.8036 43.6203 109.3018
EXTRA_LOC= -87.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVG CARTESIAN
417.6195 -1134.9092 1437.0870 8.1964 43.6203 109.3018
EXTRA_LOC= -70.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVH CARTESIAN
46.4736 -1208.4143 1437.0870 -9.8036 43.6203 109.3018
EXTRA_LOC= -88.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVI CARTESIAN
437.3628 -1127.4478 1437.0870 9.1964 43.6203 109.3018
EXTRA_LOC= -69.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVJ CARTESIAN
25.3768 -1209.0413 1437.0870 -10.8036 43.6203 109.3018
EXTRA_LOC= -89.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVK CARTESIAN
626.4976 -1034.3721 1437.0870 19.1964 43.6203 109.3018
EXTRA_LOC= -59.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVL CARTESIAN
-184.9566 -1195.0799 1437.0870 -20.8036 43.6203 109.3018
EXTRA_LOC= -99.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVM CARTESIAN
796.5965 -909.8675 1437.0870 29.1964 43.6203 109.3018
EXTRA_LOC= -49.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVN CARTESIAN
-389.6701 -1144.8066 1437.0870 -30.8036 43.6203 109.3018
EXTRA_LOC= -109.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVO CARTESIAN
942.4912 -757.7170 1437.0870 39.1964 43.6203 109.3018
EXTRA_LOC= -39.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVP CARTESIAN
-582.5437 -1059.7489 1437.0870 -40.8036 43.6203 109.3018
EXTRA_LOC= -119.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVQ CARTESIAN
1059.7489 -582.5437 1437.0870 49.1964 43.6203 109.3018
EXTRA_LOC= -29.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVR CARTESIAN
-757.7170 -942.4912 1437.0870 -50.8036 43.6203 109.3018
EXTRA_LOC= -129.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVS CARTESIAN
1144.8066 -389.6701 1437.0870 59.1964 43.6203 109.3018
EXTRA_LOC= -19.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVT CARTESIAN
-909.8675 -796.5965 1437.0870 -60.8036 43.6203 109.3018
EXTRA_LOC= -139.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;
&LOCATION: JVU CARTESIAN
1195.0799 -184.9566 1437.0870 69.1964 43.6203 109.3018
EXTRA_LOC= -9.0000 60.0000 -56.0000 10.0000 10.0000 125.0000 ;

```
&LOCATION: JVV CARTESIAN
-1034.3721 -626.4976 1437.0870 -70.8036  43.6203  109.3018
EXTRA_LOC= -149.0000  60.0000 -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JVW CARTESIAN
 1209.0413  25.3768 1437.0870  79.1964  43.6203  109.3018
EXTRA_LOC=   1.0000  60.0000 -56.0000  10.0000  10.0000  125.0000 ;
&LOCATION: JVX CARTESIAN
-1127.4478 -437.3628 1437.0870 -80.8036  43.6203  109.3018
EXTRA_LOC= -159.0000  60.0000 -56.0000  10.0000  10.0000  125.0000 ;
&END/LOCATIONS;

&BEGIN/TOOL_LOCATIONS KUKA;
&END/TOOL_LOCATIONS;

&BEGIN/PROGRAM KUKAJ1;
&COMMENT: MERGE_CELL: TIJS,WORLD;
FUNCTION=ON,HP91
WISTAT=T
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
&ACTIVE_DEVICE: KUKA;
GES ALL P1
&COMMENT: ** BEGIN_SEGMENT:STARTUP; **;
&SET_DEVICE_MOTION_MODE: INTERPOLATE;
&COMMENT: ** END_SEGMENT:STARTUP; **;
&GOTO_JOINTS: JV CARTESIAN;
&COMMENT: ;****** CTA RELEASE 7.0 *****;
&COMMENT: ** BEGIN_SEGMENT:STARTUP; **;
&SET_DEVICE_MOTION_MODE: INTERPOLATE;
&COMMENT: ** END_SEGMENT:STARTUP; **;
&GOTO_JOINTS: JV CARTESIAN;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV0 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV0 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV1 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV1 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV2 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV2 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
```

```
S A 30
&GOTO_JOINTS: JV3 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV3 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV4 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV4 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV5 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV5 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV6 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV6 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV7 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV7 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV8 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV8 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JV9 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JV9 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVA CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVA CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVB CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVB CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
```

```
S A 30
&GOTO_JOINTS: JVC CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVC CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVD CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVD CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVE CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVE CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVF CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVF CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVG CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVG CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVH CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVH CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVI CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVI CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVJ CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVJ CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVK CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVK CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
```

```
S A 30
&GOTO_JOINTS: JVL CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVL CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVM CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVM CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVN CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVN CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVO CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVO CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVP CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVP CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVQ CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVQ CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVR CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVR CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVS CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVS CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
S A 30
&GOTO_JOINTS: JVT CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: **  END_OPERATION: OUTLAY,GOTO,JVT CARTESIAN  **;
&COMMENT: **  BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN  **;
```

```
S A 30
&GOTO_JOINTS: JVU CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVU CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVV CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVV CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVW CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVW CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,GOTO,JOINTS CARTESIAN **;
S A 30
&GOTO_JOINTS: JVX CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,GOTO,JVX CARTESIAN **;
ADD P1 KON+10
WRT Z 100
JMP AD 5
&END/PROGRAM;
```

Appendix 7. The KUKA CSP-file for axis 7.

```
&BEGIN/DEVICES;
& KUKA      6 KUKA;
&END/DEVICES;

&BEGIN/LOCATIONS KUKA;
&LOCATION: TPT CARTESIAN
  320.6673 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -64.5262 39.5746 -23.3094 168.1472 -59.3314 3.2385 ;
&LOCATION: TPT1 CARTESIAN
  340.6673 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -63.1842 40.2060 -23.9396 167.7797 -59.5988 4.7017 ;
&LOCATION: TPT2 CARTESIAN
  300.6674 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -65.8988 38.9718 -22.7146 168.5331 -59.0729 1.7343 ;
&LOCATION: TPT3 CARTESIAN
  360.6672 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -61.8732 40.8650 -24.6050 167.4304 -59.8752 6.1232 ;
&LOCATION: TPT4 CARTESIAN
  280.6674 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -67.3015 38.3989 -22.1555 168.9374 -58.8232 0.1895 ;
&LOCATION: TPT5 CARTESIAN
  380.6672 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -60.5936 41.5502 -25.3053 167.0991 -60.1607 7.5028 ;
&LOCATION: TPT6 CARTESIAN
```

```
 260.6674 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -68.7335  37.8573 -21.6323 169.3601 -58.5824  -1.3948 ;
&LOCATION: TPT7 CARTESIAN
 400.6672 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -59.3453  42.2606 -26.0402 166.7855 -60.4555   8.8403 ;
&LOCATION: TPT8 CARTESIAN
 240.6675 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -70.1939  37.3482 -21.1454 169.8011 -58.3506  -3.0176 ;
&LOCATION: TPT9 CARTESIAN
 420.6671 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -58.1285  42.9949 -26.8096 166.4892 -60.7597  10.1358 ;
&LOCATION: TPT10 CARTESIAN
 220.6675 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -71.6816  36.8729 -20.6949 170.2601 -58.1278  -4.6777 ;
&LOCATION: TPT11 CARTESIAN
 440.6671 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -56.9430  43.7521 -27.6132 166.2099 -61.0735  11.3894 ;
&LOCATION: TPT12 CARTESIAN
 200.6676 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -73.1953  36.4325 -20.2813 170.7367 -57.9142  -6.3734 ;
&LOCATION: TPT13 CARTESIAN
 460.6670 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -55.7886  44.5312 -28.4509 165.9469 -61.3972  12.6014 ;
&LOCATION: TPT14 CARTESIAN
 180.6676 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -74.7336  36.0283 -19.9048 171.2306 -57.7100  -8.1030 ;
&LOCATION: TPT15 CARTESIAN
 480.6670 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -54.6649  45.3313 -29.3225 165.7000 -61.7310  13.7721 ;
&LOCATION: TPT16 CARTESIAN
 160.6676 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -76.2947  35.6613 -19.5655 171.7409 -57.5152  -9.8644 ;
&LOCATION: TPT17 CARTESIAN
 500.6670 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -53.5717  46.1514 -30.2280 165.4686 -62.0754  14.9020 ;
&LOCATION: TPT18 CARTESIAN
 140.6677 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -77.8769  35.3328 -19.2638 172.2670 -57.3303 -11.6554 ;
&LOCATION: TPT19 CARTESIAN
 520.6669 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -52.5084  46.9908 -31.1672 165.2521 -62.4305  15.9916 ;
&LOCATION: TPT20 CARTESIAN
 120.6677 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -79.4780  35.0436 -18.9998 172.8079 -57.1555 -13.4735 ;
&LOCATION: TPT21 CARTESIAN
 590.6668 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -49.0148  50.0701 -34.7207 164.6045 -63.7636  19.4971 ;
&LOCATION: TPT22 CARTESIAN
  50.6679 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -85.1985  34.3531 -18.3758 174.8015 -56.6274 -20.0089 ;
&LOCATION: TPT23 CARTESIAN
 660.6666 -731.2816 1216.2041  -61.6076   14.4223  169.0842
EXTRA_LOC= -45.8556  53.3523 -38.6926 164.1097 -65.2508  22.5465 ;
&LOCATION: TPT24 CARTESIAN
```

```
 -19.3320 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -91.0163 34.1816 -18.2221 176.9057 -56.2402 -26.7065 ;
&LOCATION: TPT25 CARTESIAN
 730.6665 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -43.0016 56.8248 -43.0973 163.7447 -66.9127 25.1727 ;
&LOCATION: TPT26 CARTESIAN
 -89.3319 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -96.8133 34.5360 -18.5402 179.0506 -56.0071 -33.4142 ;
&LOCATION: TPT27 CARTESIAN
 800.6664 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -40.4229 60.4876 -47.9630 163.4873 -68.7753 27.4078 ;
&LOCATION: TPT28 CARTESIAN
 -159.3317 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -102.4738 35.4019 -19.3271 181.1625 -55.9382 -39.9796 ;
&LOCATION: TPT29 CARTESIAN
 870.6662 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -38.0905 64.3556 -53.3378 163.3171 -70.8715 29.2805 ;
&LOCATION: TPT30 CARTESIAN
 -229.3316 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -107.8980 36.7463 -20.5757 183.1743 -56.0398 -46.2666 ;
&LOCATION: TPT31 CARTESIAN
 940.6686 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -35.9774 68.4624 -59.2995 163.2152 -73.2469 30.8133 ;
&LOCATION: TPT32 CARTESIAN
 -299.3314 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -113.0105 38.5227 -22.2757 185.0331 -56.3153 -52.1679 ;
&LOCATION: TPT33 CARTESIAN
 1010.6685 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -34.0592 72.8684 -65.9746 163.1631 -75.9682 32.0186 ;
&LOCATION: TPT34 CARTESIAN
 -369.3313 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -117.7636 40.6782 -24.4156 186.7047 -56.7667 -57.6112 ;
&LOCATION: TPT35 CARTESIAN
 1080.6683 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -32.3138 77.6815 -73.5795 163.1419 -79.1437 32.8930 ;
&LOCATION: TPT36 CARTESIAN
 -439.3337 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -122.1358 43.1601 -26.9840 188.1720 -57.3964 -62.5572 ;
&LOCATION: TPT37 CARTESIAN
 1150.6682 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -30.7217 83.1060 -82.5216 163.1280 -82.9713 33.4012 ;
&LOCATION: TPT38 CARTESIAN
 -509.3336 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -126.1260 45.9204 -29.9717 189.4327 -58.2082 -66.9942 ;
&LOCATION: TPT39 CARTESIAN
 1220.6681 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -29.2658 89.6114 -93.7357 163.0811 -87.8997 33.4257 ;
&LOCATION: TPT40 CARTESIAN
 -579.3334 -731.2816 1216.2041 -61.6076 14.4223 169.0842
EXTRA_LOC= -129.7484 48.9200 -33.3737 190.4949 -59.2082 -70.9314 ;
&END/LOCATIONS;

&BEGIN/TOOL_LOCATIONS KUKA;
&END/TOOL_LOCATIONS;
```

```
&BEGIN/PROGRAM KUKAJ7;
&COMMENT: MERGE_CELL: TIJSSTR,WORLD;
FUNCTION=ON,HP97
WISTAT=T
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
&ACTIVE_DEVICE: KUKA;
GES BAN P1
&COMMENT: ** BEGIN_SEGMENT:STARTUP; **;
&COMMENT: ** WORKING_TPOINT:KUKA06,TPW; **;
&SET_DEVICE_MOTION_MODE: INTERPOLATE;
&COMMENT: ** END_SEGMENT:STARTUP; **;
&GOTO_TPOINT: TPT CARTESIAN;
&COMMENT: ;****** PLACE RELEASE 7.0 ******;
&COMMENT: ** BEGIN_SEGMENT:STARTUP; **;
&COMMENT: ** WORKING_TPOINT:KUKA06,TPW; **;
&SET_DEVICE_MOTION_MODE: INTERPOLATE;
&COMMENT: ** END_SEGMENT:STARTUP; **;
&GOTO_TPOINT: TPT CARTESIAN;
&SET_DEVICE_MOTION_MODE: STRAIGHT;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT1 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT1 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT1 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT2 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT2 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT2 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT3 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT3 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT3 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT4 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT4 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT4 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT5 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT5 CARTESIAN;
RS A 30
WRT Z 5
```

```
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT5 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT6 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT6 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT6 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT7 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT7 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT7 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT8 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT8 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT8 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT9 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT9 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT9 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT10 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT10 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT10 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT11 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT11 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT11 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT12 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT12 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT12 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT13 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT13 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT13 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT14 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT14 CARTESIAN;
RS A 30
WRT Z 5
```

```
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT14 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT15 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT15 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT15 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT16 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT16 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT16 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT17 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT17 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT17 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT18 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT18 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT18 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT19 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT19 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT19 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT20 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT20 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT20 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT21 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT21 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT21 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT22 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT22 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT22 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT23 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT23 CARTESIAN;
RS A 30
WRT Z 5
```

```
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT23 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT24 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT24 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT24 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT25 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT25 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT25 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT26 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT26 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT26 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT27 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT27 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT27 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT28 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT28 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT28 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT29 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT29 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT29 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT30 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT30 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT30 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT31 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT31 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT31 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT32 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT32 CARTESIAN;
RS A 30
WRT Z 5
```

```
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT32 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT33 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT33 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT33 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT34 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT34 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT34 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT35 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT35 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT35 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT36 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT36 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT36 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT37 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT37 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT37 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT38 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT38 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT38 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT39 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT39 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT39 CARTESIAN **;
&COMMENT: ** BEGIN_OPERATION: OUTLAY,KUKA00,TPT40 CARTESIAN **;
S A 30
&GOTO_TPOINT: TPT40 CARTESIAN;
RS A 30
WRT Z 5
&COMMENT: ** END_OPERATION: OUTLAY,KUKA00,TPT40 CARTESIAN **;
ADD P1 KON+10
WRT Z 100
JMP AD 5
&END/PROGRAM;
```

Appendix 8. The LIS, SRC and SRL-files generated by the postprocessor.

The LIS, SRC and SRL-files for the different axis are very long and very the same. Therefor, only the LIS-file and SRL-file of axis 1 are included and for axis 1 and the straight line movement, the SRC-file is included.

The KUKAJ1.LIS file.

```
1.0   KOM ........................
2.0   KOM SRCL TRANSLATOR OUTPUT  .
3.0   KOM ........................
4.0   KOM
5.0   KOM 21.OCT.1991 17.03.09.03
6.0   KOM CSP FILE ... KUKAJ1
7.0   KOM RFILE ... KUKAJ1
8.0   KOM
9.0   KOM
10.0  KOM
11.0  KOM
12.0  DEF HP91
13.0  ORI VAR
14.0  KOM MERGE.CELL. TIJS.WORLD.
15.0  WISTAT=T
16.0  LAD P1 KON 10
17.0  LAD P2 KON 100
18.0  DEF AD 5
19.0  VGL P1 P2
20.0  BAW GR
21.0  HLT UN
22.0  KOM ACT DEVICE ... KUKA
23.0  GES ALL P1
24.0  KOM .. BEGIN.SEGMENT.STARTUP.
25.0  KOM  INTERPOLATE ... PTP SYN
26.0  KOM .. END.SEGMENT.STARTUP.
27.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
28.0  PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
29.0  KOM ....... CTA RELEASE 7.0 ..
30.0  KOM .. BEGIN.SEGMENT.STARTUP.
31.0  KOM  INTERPOLATE ... PTP SYN
32.0  KOM .. END.SEGMENT.STARTUP.
33.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
34.0  PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
35.0  KOM .. BEGIN.OPERATION. OUTLA
36.0  S A 30
37.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
38.0  PTP X+255.6 Y-1182.0 Z+1437.1 A+0.196 B+43.620 C+109.302
39.0  RS A 30
40.0  WRT Z 5
41.0  KOM .. END.OPERATION. OUTLAY.
42.0  KOM .. BEGIN.OPERATION. OUTLA
43.0  S A 30
44.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
45.0  PTP X+214.2 Y-1190.2 Z+1437.1 A-1.804 B+43.620 C+109.302
46.0  RS A 30
```

```
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+276.2 Y-1177.3 Z+1437.1 A+1.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+193.4 Y-1193.7 Z+1437.1 A-2.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+296.7 Y-1172.3 Z+1437.1 A+2.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+172.5 Y-1196.9 Z+1437.1 A-3.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+317.1 Y-1167.0 Z+1437.1 A+3.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+151.6 Y-1199.8 Z+1437.1 A-4.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+337.4 Y-1161.3 Z+1437.1 A+4.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
```

```
101.0  PTP X+130.7 Y-1202.2 Z+1437.1 A-5.804 B+43.620 C+109.302
102.0  RS A 30
103.0  WRT Z 5
104.0  KOM .. END.OPERATION. OUTLAY.
105.0  KOM .. BEGIN.OPERATION. OUTLA
106.0  S A 30
107.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
108.0  PTP X+357.7 Y-1155.2 Z+1437.1 A+5.196 B+43.620 C+109.302
109.0  RS A 30
110.0  WRT Z 5
111.0  KOM .. END.OPERATION. OUTLAY.
112.0  KOM .. BEGIN.OPERATION. OUTLA
113.0  S A 30
114.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
115.0  PTP X+109.7 Y-1204.3 Z+1437.1 A-6.804 B+43.620 C+109.302
116.0  RS A 30
117.0  WRT Z 5
118.0  KOM .. END.OPERATION. OUTLAY.
119.0  KOM .. BEGIN.OPERATION. OUTLA
120.0  S A 30
121.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
122.0  PTP X+377.8 Y-1148.8 Z+1437.1 A+6.196 B+43.620 C+109.302
123.0  RS A 30
124.0  WRT Z 5
125.0  KOM .. END.OPERATION. OUTLAY.
126.0  KOM .. BEGIN.OPERATION. OUTLA
127.0  S A 30
128.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
129.0  PTP X+88.6 Y-1206.1 Z+1437.1 A-7.804 B+43.620 C+109.302
130.0  RS A 30
131.0  WRT Z 5
132.0  KOM .. END.OPERATION. OUTLAY.
133.0  KOM .. BEGIN.OPERATION. OUTLA
134.0  S A 30
135.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
136.0  PTP X+397.7 Y-1142.0 Z+1437.1 A+7.196 B+43.620 C+109.302
137.0  RS A 30
138.0  WRT Z 5
139.0  KOM .. END.OPERATION. OUTLAY.
140.0  KOM .. BEGIN.OPERATION. OUTLA
141.0  S A 30
142.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
143.0  PTP X+67.6 Y-1207.4 Z+1437.1 A-8.804 B+43.620 C+109.302
144.0  RS A 30
145.0  WRT Z 5
146.0  KOM .. END.OPERATION. OUTLAY.
147.0  KOM .. BEGIN.OPERATION. OUTLA
148.0  S A 30
149.0  $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
150.0  PTP X+417.6 Y-1134.9 Z+1437.1 A+8.196 B+43.620 C+109.302
151.0  RS A 30
152.0  WRT Z 5
153.0  KOM .. END.OPERATION. OUTLAY.
154.0  KOM .. BEGIN.OPERATION. OUTLA
```

```
155.0   S A 30
156.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
157.0   PTP X+46.5 Y-1208.4 Z+1437.1 A-9.804 B+43.620 C+109.302
158.0   RS A 30
159.0   WRT Z 5
160.0   KOM .. END.OPERATION. OUTLAY.
161.0   KOM .. BEGIN.OPERATION. OUTLA
162.0   S A 30
163.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
164.0   PTP X+437.4 Y-1127.4 Z+1437.1 A+9.196 B+43.620 C+109.302
165.0   RS A 30
166.0   WRT Z 5
167.0   KOM .. END.OPERATION. OUTLAY.
168.0   KOM .. BEGIN.OPERATION. OUTLA
169.0   S A 30
170.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
171.0   PTP X+25.4 Y-1209.0 Z+1437.1 A-10.804 B+43.620 C+109.302
172.0   RS A 30
173.0   WRT Z 5
174.0   KOM .. END.OPERATION. OUTLAY.
175.0   KOM .. BEGIN.OPERATION. OUTLA
176.0   S A 30
177.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
178.0   PTP X+626.5 Y-1034.4 Z+1437.1 A+19.196 B+43.620 C+109.302
179.0   RS A 30
180.0   WRT Z 5
181.0   KOM .. END.OPERATION. OUTLAY.
182.0   KOM .. BEGIN.OPERATION. OUTLA
183.0   S A 30
184.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
185.0   PTP X-185.0 Y-1195.1 Z+1437.1 A-20.804 B+43.620 C+109.302
186.0   RS A 30
187.0   WRT Z 5
188.0   KOM .. END.OPERATION. OUTLAY.
189.0   KOM .. BEGIN.OPERATION. OUTLA
190.0   S A 30
191.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
192.0   PTP X+796.6 Y-909.9 Z+1437.1 A+29.196 B+43.620 C+109.302
193.0   RS A 30
194.0   WRT Z 5
195.0   KOM .. END.OPERATION. OUTLAY.
196.0   KOM .. BEGIN.OPERATION. OUTLA
197.0   S A 30
198.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
199.0   PTP X-389.7 Y-1144.8 Z+1437.1 A-30.804 B+43.620 C+109.302
200.0   RS A 30
201.0   WRT Z 5
202.0   KOM .. END.OPERATION. OUTLAY.
203.0   KOM .. BEGIN.OPERATION. OUTLA
204.0   S A 30
205.0   $WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
206.0   PTP X+942.5 Y-757.7 Z+1437.1 A+39.196 B+43.620 C+109.302
207.0   RS A 30
208.0   WRT Z 5
```

```
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-582.5 Y-1059.7 Z+1437.1 A-40.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1059.7 Y-582.5 Z+1437.1 A+49.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-757.7 Y-942.5 Z+1437.1 A-50.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1144.8 Y-389.7 Z+1437.1 A+59.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-909.9 Y-796.6 Z+1437.1 A-60.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1195.1 Y-185.0 Z+1437.1 A+69.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-1034.4 Y-626.5 Z+1437.1 A-70.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1P 2P 3N 4P 5P 6P)
PTP X+1209.0 Y+25.4 Z+1437.1 A+79.196 B+43.620 C+109.302
```

```
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-1127.4 Y-437.4 Z+1437.1 A-80.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
ADD P1 KON+10
WRT Z 100
JMP AD 5
END HP91
TOTAL NUMBER OF ERRORS THIS COMPILATION: 0
TOTAL NUMBER OF WARNINGS THIS COMPILATION: 0
```

Appendix 8. The KUKAJ1.SRL file.

```
00001,22100,2,0,1,HP91;
00002,28000,ORI VAR;
00003,28000,KOM MERGE.CELL. TIJS.WORLD.;
00004,28000,LAD P1 KON 10;
00005,28000,LAD P2 KON 100;
00006,28000,DEF AD 5;
00007,28000,VGL P1 P2;
00008,28000,BAW GR;
00009,28000,HLT UN;
00010,28000,GES ALL P1;
00011,28000,KOM .. BEGIN.SEGMENT.STARTUP.;
00012,28000,KOM .. END.SEGMENT.STARTUP. ;
00013,28100, 5;
00014,28000,PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302;
00015,28000,KOM ....... CTA RELEASE 7.0 ..;
00016,28000,KOM .. BEGIN.SEGMENT.STARTUP.;
00017,28000,KOM .. END.SEGMENT.STARTUP. ;
00018,28100, 5;
00019,28000,PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302;
00020,28000,KOM .. BEGIN.OPERATION. OUTLA;
00021,28000,S A 30 ;
00022,28100, 5;
00023,28000,PTP X+255.6 Y-1182.0 Z+1437.1 A+0.196 B+43.620 C+109.302;
00024,28000,RS A 30 ;
00025,28000,WRT Z 5 ;
00026,28000,KOM .. END.OPERATION. OUTLAY.;
00027,28000,KOM .. BEGIN.OPERATION. OUTLA;
00028,28000,S A 30 ;
00029,28100, 5;
00030,28000,PTP X+214.2 Y-1190.2 Z+1437.1 A-1.804 B+43.620 C+109.302;
00031,28000,RS A 30 ;
00032,28000,WRT Z 5 ;
00033,28000,KOM .. END.OPERATION. OUTLAY.;
00034,28000,KOM .. BEGIN.OPERATION. OUTLA;
```

```
00035,28000,S A 30 ;
00036,28100, 5;
00037,28000,PTP X+276.2 Y-1177.3 Z+1437.1 A+1.196 B+43.620 C+109.302;
00038,28000,RS A 30 ;
00039,28000,WRT Z 5 ;
00040,28000,KOM .. END.OPERATION. OUTLAY.;
00041,28000,KOM .. BEGIN.OPERATION. OUTLA;
00042,28000,S A 30 ;
00043,28100, 5;
00044,28000,PTP X+193.4 Y-1193.7 Z+1437.1 A-2.804 B+43.620 C+109.302;
00045,28000,RS A 30 ;
00046,28000,WRT Z 5 ;
00047,28000,KOM .. END.OPERATION. OUTLAY.;
00048,28000,KOM .. BEGIN.OPERATION. OUTLA;
00049,28000,S A 30 ;
00050,28100, 5;
00051,28000,PTP X+296.7 Y-1172.3 Z+1437.1 A+2.196 B+43.620 C+109.302;
00052,28000,RS A 30 ;
00053,28000,WRT Z 5 ;
00054,28000,KOM .. END.OPERATION. OUTLAY.;
00055,28000,KOM .. BEGIN.OPERATION. OUTLA;
00056,28000,S A 30 ;
00057,28100, 5;
00058,28000,PTP X+172.5 Y-1196.9 Z+1437.1 A-3.804 B+43.620 C+109.302;
00059,28000,RS A 30 ;
00060,28000,WRT Z 5 ;
00061,28000,KOM .. END.OPERATION. OUTLAY.;
00062,28000,KOM .. BEGIN.OPERATION. OUTLA;
00063,28000,S A 30 ;
00064,28100, 5;
00065,28000,PTP X+317.1 Y-1167.0 Z+1437.1 A+3.196 B+43.620 C+109.302;
00066,28000,RS A 30 ;
00067,28000,WRT Z 5 ;
00068,28000,KOM .. END.OPERATION. OUTLAY.;
00069,28000,KOM .. BEGIN.OPERATION. OUTLA;
00070,28000,S A 30 ;
00071,28100, 5;
00072,28000,PTP X+151.6 Y-1199.8 Z+1437.1 A-4.804 B+43.620 C+109.302;
00073,28000,RS A 30 ;
00074,28000,WRT Z 5 ;
00075,28000,KOM .. END.OPERATION. OUTLAY.;
00076,28000,KOM .. BEGIN.OPERATION. OUTLA;
00077,28000,S A 30 ;
00078,28100, 5;
00079,28000,PTP X+337.4 Y-1161.3 Z+1437.1 A+4.196 B+43.620 C+109.302;
00080,28000,RS A 30 ;
00081,28000,WRT Z 5 ;
00082,28000,KOM .. END.OPERATION. OUTLAY.;
00083,28000,KOM .. BEGIN.OPERATION. OUTLA;
00084,28000,S A 30 ;
00085,28100, 5;
00086,28000,PTP X+130.7 Y-1202.2 Z+1437.1 A-5.804 B+43.620 C+109.302;
00087,28000,RS A 30 ;
00088,28000,WRT Z 5 ;
```

```
00089,28000,KOM .. END.OPERATION. OUTLAY.;
00090,28000,KOM .. BEGIN.OPERATION. OUTLA;
00091,28000,S A 30 ;
00092,28100, 5;
00093,28000,PTP X+357.7 Y-1155.2 Z+1437.1 A+5.196 B+43.620 C+109.302;
00094,28000,RS A 30 ;
00095,28000,WRT Z 5 ;
00096,28000,KOM .. END.OPERATION. OUTLAY.;
00097,28000,KOM .. BEGIN.OPERATION. OUTLA;
00098,28000,S A 30 ;
00099,28100, 5;
00100,28000,PTP X+109.7 Y-1204.3 Z+1437.1 A-6.804 B+43.620 C+109.302;
00101,28000,RS A 30 ;
00102,28000,WRT Z 5 ;
00103,28000,KOM .. END.OPERATION. OUTLAY.;
00104,28000,KOM .. BEGIN.OPERATION. OUTLA;
00105,28000,S A 30 ;
00106,28100, 5;
00107,28000,PTP X+377.8 Y-1148.8 Z+1437.1 A+6.196 B+43.620 C+109.302;
00108,28000,RS A 30 ;
00109,28000,WRT Z 5 ;
00110,28000,KOM .. END.OPERATION. OUTLAY.;
00111,28000,KOM .. BEGIN.OPERATION. OUTLA;
00112,28000,S A 30 ;
00113,28100, 5;
00114,28000,PTP X+88.6 Y-1206.1 Z+1437.1 A-7.804 B+43.620 C+109.302;
00115,28000,RS A 30 ;
00116,28000,WRT Z 5 ;
00117,28000,KOM .. END.OPERATION. OUTLAY.;
00118,28000,KOM .. BEGIN.OPERATION. OUTLA;
00119,28000,S A 30 ;
00120,28100, 5;
00121,28000,PTP X+397.7 Y-1142.0 Z+1437.1 A+7.196 B+43.620 C+109.302;
00122,28000,RS A 30 ;
00123,28000,WRT Z 5 ;
00124,28000,KOM .. END.OPERATION. OUTLAY.;
00125,28000,KOM .. BEGIN.OPERATION. OUTLA;
00126,28000,S A 30 ;
00127,28100, 5;
00128,28000,PTP X+67.6 Y-1207.4 Z+1437.1 A-8.804 B+43.620 C+109.302;
00129,28000,RS A 30 ;
00130,28000,WRT Z 5 ;
00131,28000,KOM .. END.OPERATION. OUTLAY.;
00132,28000,KOM .. BEGIN.OPERATION. OUTLA;
00133,28000,S A 30 ;
00134,28100, 5;
00135,28000,PTP X+417.6 Y-1134.9 Z+1437.1 A+8.196 B+43.620 C+109.302;
00136,28000,RS A 30 ;
00137,28000,WRT Z 5 ;
00138,28000,KOM .. END.OPERATION. OUTLAY.;
00139,28000,KOM .. BEGIN.OPERATION. OUTLA;
00140,28000,S A 30 ;
00141,28100, 5;
00142,28000,PTP X+46.5 Y-1208.4 Z+1437.1 A-9.804 B+43.620 C+109.302;
```

```
00143,28000,RS A 30 ;
00144,28000,WRT Z 5 ;
00145,28000,KOM .. END.OPERATION. OUTLAY.;
00146,28000,KOM .. BEGIN.OPERATION. OUTLA;
00147,28000,S A 30 ;
00148,28100, 5;
00149,28000,PTP X+437.4 Y-1127.4 Z+1437.1 A+9.196 B+43.620 C+109.302;
00150,28000,RS A 30 ;
00151,28000,WRT Z 5 ;
00152,28000,KOM .. END.OPERATION. OUTLAY.;
00153,28000,KOM .. BEGIN.OPERATION. OUTLA;
00154,28000,S A 30 ;
00155,28100, 5;
00156,28000,PTP X+25.4 Y-1209.0 Z+1437.1 A-10.804 B+43.620 C+109.302;
00157,28000,RS A 30 ;
00158,28000,WRT Z 5 ;
00159,28000,KOM .. END.OPERATION. OUTLAY.;
00160,28000,KOM .. BEGIN.OPERATION. OUTLA;
00161,28000,S A 30 ;
00162,28100, 5;
00163,28000,PTP X+626.5 Y-1034.4 Z+1437.1 A+19.196 B+43.620 C+109.302;
00164,28000,RS A 30 ;
00165,28000,WRT Z 5 ;
00166,28000,KOM .. END.OPERATION. OUTLAY.;
00167,28000,KOM .. BEGIN.OPERATION. OUTLA;
00168,28000,S A 30 ;
00169,28100, 5;
00170,28000,PTP X-185.0 Y-1195.1 Z+1437.1 A-20.804 B+43.620 C+109.302;
00171,28000,RS A 30 ;
00172,28000,WRT Z 5 ;
00173,28000,KOM .. END.OPERATION. OUTLAY.;
00174,28000,KOM .. BEGIN.OPERATION. OUTLA;
00175,28000,S A 30 ;
00176,28100, 5;
00177,28000,PTP X+796.6 Y-909.9 Z+1437.1 A+29.196 B+43.620 C+109.302;
00178,28000,RS A 30 ;
00179,28000,WRT Z 5 ;
00180,28000,KOM .. END.OPERATION. OUTLAY.;
00181,28000,KOM .. BEGIN.OPERATION. OUTLA;
00182,28000,S A 30 ;
00183,28100, 5;
00184,28000,PTP X-389.7 Y-1144.8 Z+1437.1 A-30.804 B+43.620 C+109.302;
00185,28000,RS A 30 ;
00186,28000,WRT Z 5 ;
00187,28000,KOM .. END.OPERATION. OUTLAY.;
00188,28000,KOM .. BEGIN.OPERATION. OUTLA;
00189,28000,S A 30 ;
00190,28100, 5;
00191,28000,PTP X+942.5 Y-757.7 Z+1437.1 A+39.196 B+43.620 C+109.302;
00192,28000,RS A 30 ;
00193,28000,WRT Z 5 ;
00194,28000,KOM .. END.OPERATION. OUTLAY.;
00195,28000,KOM .. BEGIN.OPERATION. OUTLA;
00196,28000,S A 30 ;
```

```
00197,28100, 5;
00198,28000,PTP X-582.5 Y-1059.7 Z+1437.1 A-40.804 B+43.620 C+109.302;
00199,28000,RS A 30 ;
00200,28000,WRT Z 5 ;
00201,28000,KOM .. END.OPERATION. OUTLAY.;
00202,28000,KOM .. BEGIN.OPERATION. OUTLA;
00203,28000,S A 30 ;
00204,28100, 5;
00205,28000,PTP X+1059.7 Y-582.5 Z+1437.1 A+49.196 B+43.620 C+109.302;
00206,28000,RS A 30 ;
00207,28000,WRT Z 5 ;
00208,28000,KOM .. END.OPERATION. OUTLAY.;
00209,28000,KOM .. BEGIN.OPERATION. OUTLA;
00210,28000,S A 30 ;
00211,28100, 5;
00212,28000,PTP X-757.7 Y-942.5 Z+1437.1 A-50.804 B+43.620 C+109.302;
00213,28000,RS A 30 ;
00214,28000,WRT Z 5 ;
00215,28000,KOM .. END.OPERATION. OUTLAY.;
00216,28000,KOM .. BEGIN.OPERATION. OUTLA;
00217,28000,S A 30 ;
00218,28100, 5;
00219,28000,PTP X+1144.8 Y-389.7 Z+1437.1 A+59.196 B+43.620 C+109.302;
00220,28000,RS A 30 ;
00221,28000,WRT Z 5 ;
00222,28000,KOM .. END.OPERATION. OUTLAY.;
00223,28000,KOM .. BEGIN.OPERATION. OUTLA;
00224,28000,S A 30 ;
00225,28100, 5;
00226,28000,PTP X-909.9 Y-796.6 Z+1437.1 A-60.804 B+43.620 C+109.302;
00227,28000,RS A 30 ;
00228,28000,WRT Z 5 ;
00229,28000,KOM .. END.OPERATION. OUTLAY.;
00230,28000,KOM .. BEGIN.OPERATION. OUTLA;
00231,28000,S A 30 ;
00232,28100, 5;
00233,28000,PTP X+1195.1 Y-185.0 Z+1437.1 A+69.196 B+43.620 C+109.302;
00234,28000,RS A 30 ;
00235,28000,WRT Z 5 ;
00236,28000,KOM .. END.OPERATION. OUTLAY.;
00237,28000,KOM .. BEGIN.OPERATION. OUTLA;
00238,28000,S A 30 ;
00239,28100, 5;
00240,28000,PTP X-1034.4 Y-626.5 Z+1437.1 A-70.804 B+43.620 C+109.302;
00241,28000,RS A 30 ;
00242,28000,WRT Z 5 ;
00243,28000,KOM .. END.OPERATION. OUTLAY.;
00244,28000,KOM .. BEGIN.OPERATION. OUTLA;
00245,28000,S A 30 ;
00246,28100, 4;
00247,28000,PTP X+1209.0 Y+25.4 Z+1437.1 A+79.196 B+43.620 C+109.302;
00248,28000,RS A 30 ;
00249,28000,WRT Z 5 ;
00250,28000,KOM .. END.OPERATION. OUTLAY.;
```

```
00251,28000,KOM ..  BEGIN.OPERATION. OUTLA;
00252,28000,S A 30 ;
00253,28100,  5;
00254,28000,PTP X-1127.4 Y-437.4 Z+1437.1 A-80.804 B+43.620 C+109.302;
00255,28000,RS A 30 ;
00256,28000,WRT Z 5 ;
00257,28000,KOM ..  END.OPERATION. OUTLAY.;
00258,28000,ADD P1 KON+10;
00259,28000,WRT Z 100;
00260,28000,JMP AD 5;
00261,28000,END HP91;
00262,22150;
```

Appendix 8. The KUKAJ1.SRC file.

```
KOM .........................
KOM SRCL TRANSLATOR OUTPUT  .
KOM .........................
KOM
KOM 21.OCT.1991 17.03.09.03
KOM CSP FILE ... KUKAJ1
KOM RFILE ... KUKAJ1
KOM
KOM
KOM
KOM
DEF HP91
ORI VAR
KOM MERGE.CELL. TIJS.WORLD.
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
KOM ACT DEVICE ... KUKA
GES ALL P1
KOM ..  BEGIN.SEGMENT.STARTUP.
KOM  INTERPOLATE ... PTP SYN
KOM ..  END.SEGMENT.STARTUP.
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
KOM ....... CTA RELEASE 7.0 ..
KOM ..  BEGIN.SEGMENT.STARTUP.
KOM  INTERPOLATE ... PTP SYN
KOM ..  END.SEGMENT.STARTUP.
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
```

```
PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+255.6 Y-1182.0 Z+1437.1 A+0.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+214.2 Y-1190.2 Z+1437.1 A-1.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+276.2 Y-1177.3 Z+1437.1 A+1.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+193.4 Y-1193.7 Z+1437.1 A-2.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+296.7 Y-1172.3 Z+1437.1 A+2.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+172.5 Y-1196.9 Z+1437.1 A-3.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+317.1 Y-1167.0 Z+1437.1 A+3.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+151.6 Y-1199.8 Z+1437.1 A-4.804 B+43.620 C+109.302
```

```
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+337.4 Y-1161.3 Z+1437.1 A+4.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+130.7 Y-1202.2 Z+1437.1 A-5.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+357.7 Y-1155.2 Z+1437.1 A+5.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+109.7 Y-1204.3 Z+1437.1 A-6.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+377.8 Y-1148.8 Z+1437.1 A+6.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+88.6 Y-1206.1 Z+1437.1 A-7.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+397.7 Y-1142.0 Z+1437.1 A+7.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
```

```
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+67.6 Y-1207.4 Z+1437.1 A-8.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+417.6 Y-1134.9 Z+1437.1 A+8.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+46.5 Y-1208.4 Z+1437.1 A-9.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+437.4 Y-1127.4 Z+1437.1 A+9.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+25.4 Y-1209.0 Z+1437.1 A-10.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+626.5 Y-1034.4 Z+1437.1 A+19.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-185.0 Y-1195.1 Z+1437.1 A-20.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+796.6 Y-909.9 Z+1437.1 A+29.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
```

```
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-389.7 Y-1144.8 Z+1437.1 A-30.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+942.5 Y-757.7 Z+1437.1 A+39.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-582.5 Y-1059.7 Z+1437.1 A-40.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1059.7 Y-582.5 Z+1437.1 A+49.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-757.7 Y-942.5 Z+1437.1 A-50.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1144.8 Y-389.7 Z+1437.1 A+59.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-909.9 Y-796.6 Z+1437.1 A-60.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X+1195.1 Y-185.0 Z+1437.1 A+69.196 B+43.620 C+109.302
RS A 30
```

```
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-1034.4 Y-626.5 Z+1437.1 A-70.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1P 2P 3N 4P 5P 6P)
PTP X+1209.0 Y+25.4 Z+1437.1 A+79.196 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5P 6P)
PTP X-1127.4 Y-437.4 Z+1437.1 A-80.804 B+43.620 C+109.302
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
ADD P1 KON+10
WRT Z 100
JMP AD 5
END HP91
```

Appendix 8. The KUKAJ7.SRC file.

```
KOM .........................
KOM SRCL TRANSLATOR OUTPUT .
KOM .........................
KOM
KOM 21.OCT.1991 17.07.58.83
KOM CSP FILE ... KUKAJ7
KOM RFILE ... KUKAJ7
KOM
KOM
KOM
KOM
DEF HP97
ORI VAR
KOM MERGE.CELL. TIJSSTR.WORLD.
LAD P1 KON 10
LAD P2 KON 100
DEF AD 5
VGL P1 P2
BAW GR
HLT UN
KOM ACT DEVICE ... KUKA
GES BAN P1
```

```
KOM .. BEGIN.SEGMENT.STARTUP.
KOM .. WORKING.TPOINT.KUKA06.
KOM INTERPOLATE ... PTP SYN
KOM .. END.SEGMENT.STARTUP.
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
PTP X+320.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
KOM ....... PLACE RELEASE 7.0
KOM .. BEGIN.SEGMENT.STARTUP.
KOM .. WORKING.TPOINT.KUKA06.
KOM INTERPOLATE ... PTP SYN
KOM .. END.SEGMENT.STARTUP.
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
PTP X+320.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
KOM STRAIGHT ... 3D LINEAR
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+340.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+300.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+360.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+280.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+380.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+260.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
```

```
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+400.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+240.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+420.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+220.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+440.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+200.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+460.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
```

```
LIN X+180.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+480.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+160.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+500.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+140.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+520.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+120.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+590.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
```

```
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X+50.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+660.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-19.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+730.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-89.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+800.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-159.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+870.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
```

```
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-229.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+940.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-299.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+1010.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-369.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+1080.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-439.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+1150.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
```

```
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-509.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6P)
LIN X+1220.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
S A 30
$WISTAT_T(BAA 1N 2P 3N 4P 5N 6N)
LIN X-579.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
RS A 30
WRT Z 5
KOM .. END.OPERATION. OUTLAY.
ADD P1 KON+10
WRT Z 100
JMP AD 5
END HP97
```

Appendix 9. Translation (german-dutch) file for the VAX.

For the VAX-VMS system a translation program is written. It is a very simple program due to the fact that the translation consists of simple ASCII-transformations. The translation program is for every axis the same, except for the filenames and the file length. The translation programs must have a filename followed by the extension COM, if you want them to be executable. You can run them on the VAX by typing @FILENAME (without extension).

```
EDIT [UG_USERS.USERNAME]FILENAME.SRC
SUBSTITUTE/HLT UN/HLT OV/1:300
SUBSTITUTE/GES/SNH/1:300
SUBSTITUTE/RS A/TZ U/1:300
SUBSTITUTE/S A/Z U/1:300
SUBSTITUTE/BAW/VIN/1:300
SUBSTITUTE/JMP/SPG/1:300
SUBSTITUTE/ADD/OPT/1:300
EXIT HP91/SAVE
EXIT
```

Figure A9.1. TIJS.COM, translation program.

D ↓ N

A	Ausgang	U	uitgang
ACH	eine Achse PTP	AS	een as bij PTP
AD	Adresse	AD	adres
ADD	Addition	OPT	optellen
ALL	alle Achsen PTP	ALL	alle assen bij PTP
ANF	Anfang	IN	inschakelen
ARI	Arithmetik Anweisung	ARI	arithmetische instruktie
AUF	oefnen	OP	openen
AUS	Ausschalten	UIT	uitschakelen
B	Bitspeicher	B	bitgeheugen op "1"
BAN	Bahn	BAN	bij baansturing
BAW	Bedingte Anweisung	VIN	voorwaardelijke instructie
BE	Bedingt	VW	voorwaardelijk
BEA	Bearbeiten	AFW	afwerken
BES	Beschleunigungs Anweisung	VSN	versnellings instruktie
BS	Bandsynchronisation Anweisung	BS	bandsynchronisatie instruktie
DEF	Definitions Anweisung	DEF	definitie instruktie
DIV	Division	DEL	delen
E	Eingang	I	ingang
EIN	Einschalten	IN	inschakelen
ESP	Eingangsspeicher	IGH	ingave geheugen
F	Falsch	V	boleaans vals?
GES	Geschwindigkeits Anweisung	SNH	snelheids instruktie
GL	Gleich 0?	GL	gelijk aan 0?
GR	groesser 0?	GR	groter dan 0?
GRF	Greifer Anweisung	GRP	grijper instruktie
HLT	Halte Anweisung	HLT	(geprogrammeerde) halt instruktie
HP	Hauptprogramm	HP	hoofdprogramma
IA	Impulsausgang	IU	impulsuitgang
KL	Kleiner 0?	KL	kleiner dan 0?
KON	Konstant	KON	konstant
LAD	Lade Anweisung	LAD	laad instruktie
LIN	Lineare Bewegungs Anweisung	LIN	lineaire bewegings instruktie
M	Merker	T	tussengeheugen op "1"
MLT	Multiplikation	VRM	vermenigvuldigen
MW	M-Wort	MW	parameterwoord M
NB	Nicht-Bitspeicher	NB	bitgeheugen op "0"
NBE	Nicht Bearbeiten	NAF	niet afwerken
NE	Nicht-Eingang	NI	niet-ingang
NM	Nicht-Merker	NT	tussengeheugen op "0"
NOP	Leerbefehl	NUL	nuloperatie instruktie
NPK	Nullpunktkorrektur Anweisung	NPK	nulpuntkorrektie instruktie
NR	Tafelnummer	NR	tabelnummer
O	ODER Anweisung	O	boleaanse OF instruktie
ORI	Orientierungs Anweisung	ORI	orienterings instruktie
OV	Override	OV	overloop
P	Parameter	P	parameter
PAU	Peripherie Ausgabe Anweisung	PUI	periferie uitgave instruktie
PND	Pendeln Anweisung	PND	pendelen instruktie
POS	Aktuelle Position laden	AAN	ingave door aanleren
POS	Ist-position uebernehmen	POS	positie overnemen
PRG	laden per Programm	NUM	numerische ingave
PTP	PTP Bewegungs Anweisung	PTP	punt tot punt bewegings instr.
PW	P-Wort	PW	parameterwoord P
RDL	Restdurchlaufzahl loeschen	RAU	rest-doorloop-aantal uitwissen
RS	Ruecksetz Anweisung	TZ	terugzet instruktie
S	Setz Anweisung	Z	zet instruktie
SF	Sensorfunktions Anweisung	SF	sensorfunktie instruktie
SPG	Sprung Anweisung	SPG	sprong instruktie
SUB	Subtraktion	AFT	aftrekken
TV	Transferieren Anweisung	VO	boleaanse verbindings instr.
TXT	Textausgabe Anweisung	TXT	tekst uitgave instruktie
U	UND Anweisung	E	boleaanse EN instruktie
UES	Ueberschleif Anweisung	LUW	luswerkings instruktie
UG	Ungleich 0?	OG	ongelijk aan 0?
UN	Unbedingt	OV	onvoorwaardelijk
UNT	Unterbrechungs Anweisung	OND	onderbrekings instruktie
UNT	Unterbrechen	OND	onderbreken
UP	Unterprogramm	OP	onderprogramma
VAR	Variabel	VAR	variabel
VGL	Vergleich	VGL	vergelijken
VSP	Variabelenspeicher	VGH	variabelen geheugen
VZ	Verzweigung	AT	aftakking
W	Wahr?	W	boleaans waar?
WRT	Warte Anweisung	WCH	wachten instruktie
WZK	Werkzeugkorrektur Anweisung	WTK	werktuigkorrektie instruktie
Z	Zeit	T	tijd in 0.1 s
ZU	Schliessen	SL	sluiten
ZY	Zyklus	CY	cyclus

Appendix 10. The SRC-files (dutch).

Only the dutch SRC-files for axis 1 and the straight line are included, due to the fact that all the SRC-files are very long and alike.

The HP91 file.

```
KOM .........................
KOM SRCL TRANSLATOR OUTPUT  .
KOM .........................
KOM
KOM 21.OCT.1991 17.03.09.03
KOM CSP FILE ... KUKAJ1
KOM RFILE ... KUKAJ1
KOM
KOM
KOM
KOM
DEF HP91
ORI VAR
KOM MERGE.CELL. TIJS.WORLD.
LAD P1 KON 10
LAD P2 KON 100
KOM DEF AD 5
VGL P1 P2
VIN GR
HLT OV
KOM ACT DEVICE ... KUKA
SNH ALL P1
KOM ..  BEGIN.SEGMENT.STARTUP.
KOM  INTERPOLATE ... PTP SYN
KOM ..  END.SEGMENT.STARTUP.
$WISTAT_D(0)
PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
KOM ....... CTA RELEASE 7.0 ..
KOM ..  BEGIN.SEGMENT.STARTUP.
KOM  INTERPOLATE ... PTP SYN
KOM ..  END.SEGMENT.STARTUP.
$WISTAT_D(0)
PTP X+234.9 Y-1186.3 Z+1437.1 A-0.804 B+43.620 C+109.302
KOM ..  BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+255.6 Y-1182.0 Z+1437.1 A+0.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM ..  END.OPERATION. OUTLAY.
KOM ..  BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+214.2 Y-1190.2 Z+1437.1 A-1.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM ..  END.OPERATION. OUTLAY.
```

```
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+276.2 Y-1177.3 Z+1437.1 A+1.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+193.4 Y-1193.7 Z+1437.1 A-2.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+296.7 Y-1172.3 Z+1437.1 A+2.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+172.5 Y-1196.9 Z+1437.1 A-3.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+317.1 Y-1167.0 Z+1437.1 A+3.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+151.6 Y-1199.8 Z+1437.1 A-4.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+337.4 Y-1161.3 Z+1437.1 A+4.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+130.7 Y-1202.2 Z+1437.1 A-5.804 B+43.620 C+109.302
TZ U 30
```

```
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+357.7 Y-1155.2 Z+1437.1 A+5.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+109.7 Y-1204.3 Z+1437.1 A-6.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+377.8 Y-1148.8 Z+1437.1 A+6.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+88.6 Y-1206.1 Z+1437.1 A-7.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+397.7 Y-1142.0 Z+1437.1 A+7.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+67.6 Y-1207.4 Z+1437.1 A-8.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+417.6 Y-1134.9 Z+1437.1 A+8.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
```

```
PTP X+46.5 Y-1208.4 Z+1437.1 A-9.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+437.4 Y-1127.4 Z+1437.1 A+9.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+25.4 Y-1209.0 Z+1437.1 A-10.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+626.5 Y-1034.4 Z+1437.1 A+19.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-185.0 Y-1195.1 Z+1437.1 A-20.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+796.6 Y-909.9 Z+1437.1 A+29.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-389.7 Y-1144.8 Z+1437.1 A-30.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+942.5 Y-757.7 Z+1437.1 A+39.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
```

```
Z U 30
$WISTAT_D(0)
PTP X-582.5 Y-1059.7 Z+1437.1 A-40.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+1059.7 Y-582.5 Z+1437.1 A+49.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-757.7 Y-942.5 Z+1437.1 A-50.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+1144.8 Y-389.7 Z+1437.1 A+59.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-909.9 Y-796.6 Z+1437.1 A-60.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+1195.1 Y-185.0 Z+1437.1 A+69.196 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-1034.4 Y-626.5 Z+1437.1 A-70.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X+1209.0 Y+25.4 Z+1437.1 A+79.196 B+43.620 C+109.302
TZ U 30
WCH T 5
```

```
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(0)
PTP X-1127.4 Y-437.4 Z+1437.1 A-80.804 B+43.620 C+109.302
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
OPT P1 KON+10
WCH T 100
SPG AD 5
END HP91
```

Appendix 10. The HP97 file.

```
KOM .........................
KOM SRCL TRANSLATOR OUTPUT .
KOM .........................
KOM
KOM 21.OCT.1991 17.07.58.83
KOM CSP FILE ... KUKAJ7
KOM RFILE ... KUKAJ7
KOM
KOM
KOM
KOM
DEF HP97
ORI VAR
KOM MERGE.CELL. TIJSSTR.WORLD.
LAD P1 KON 10
LAD P2 KON 100
KOM DEF AD 5
VGL P1 P2
VIN GR
HLT OV
KOM ACT DEVICE ... KUKA
SNH BAN P1
KOM .. BEGIN.SEGMENT.STARTUP.
KOM .. WORKING.TPOINT.KUKA06.
KOM INTERPOLATE ... PTP SYN
KOM .. END.SEGMENT.STARTUP.
$WISTAT_D(24)
PTP X+320.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
KOM ....... PLACE RELEASE 7.0
KOM .. BEGIN.SEGMENT.STARTUP.
KOM .. WORKING.TPOINT.KUKA06.
KOM INTERPOLATE ... PTP SYN
KOM .. END.SEGMENT.STARTUP.
$WISTAT_D(24)
PTP X+320.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
KOM STRAIGHT ... 3D LINEAR
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
```

```
LIN X+340.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+300.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+360.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+280.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+380.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+260.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+400.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+240.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
```

```
Z U 30
$WISTAT_D(24)
LIN X+420.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+220.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+440.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+200.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+460.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+180.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+480.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+160.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
```

```
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+500.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+140.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+520.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+120.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+590.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X+50.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+660.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-19.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
```

```
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+730.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-89.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+800.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-159.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+870.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-229.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+940.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
```

```
$WISTAT_D(56)
LIN X-299.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+1010.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-369.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+1080.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-439.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+1150.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-509.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(24)
LIN X+1220.7 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
```

```
KOM .. BEGIN.OPERATION. OUTLA
Z U 30
$WISTAT_D(56)
LIN X-579.3 Y-731.3 Z+1216.2 A-61.608 B+14.422 C+169.084
TZ U 30
WCH T 5
KOM .. END.OPERATION. OUTLAY.
OPT P1 KON+10
WCH T 100
SPG AD 5
END HP97
```

Appendix 11. Robot Controller output port 30.

The Robot Controller output port 30 is located in connection X13. The pins 2z2 and 2z28 are used.

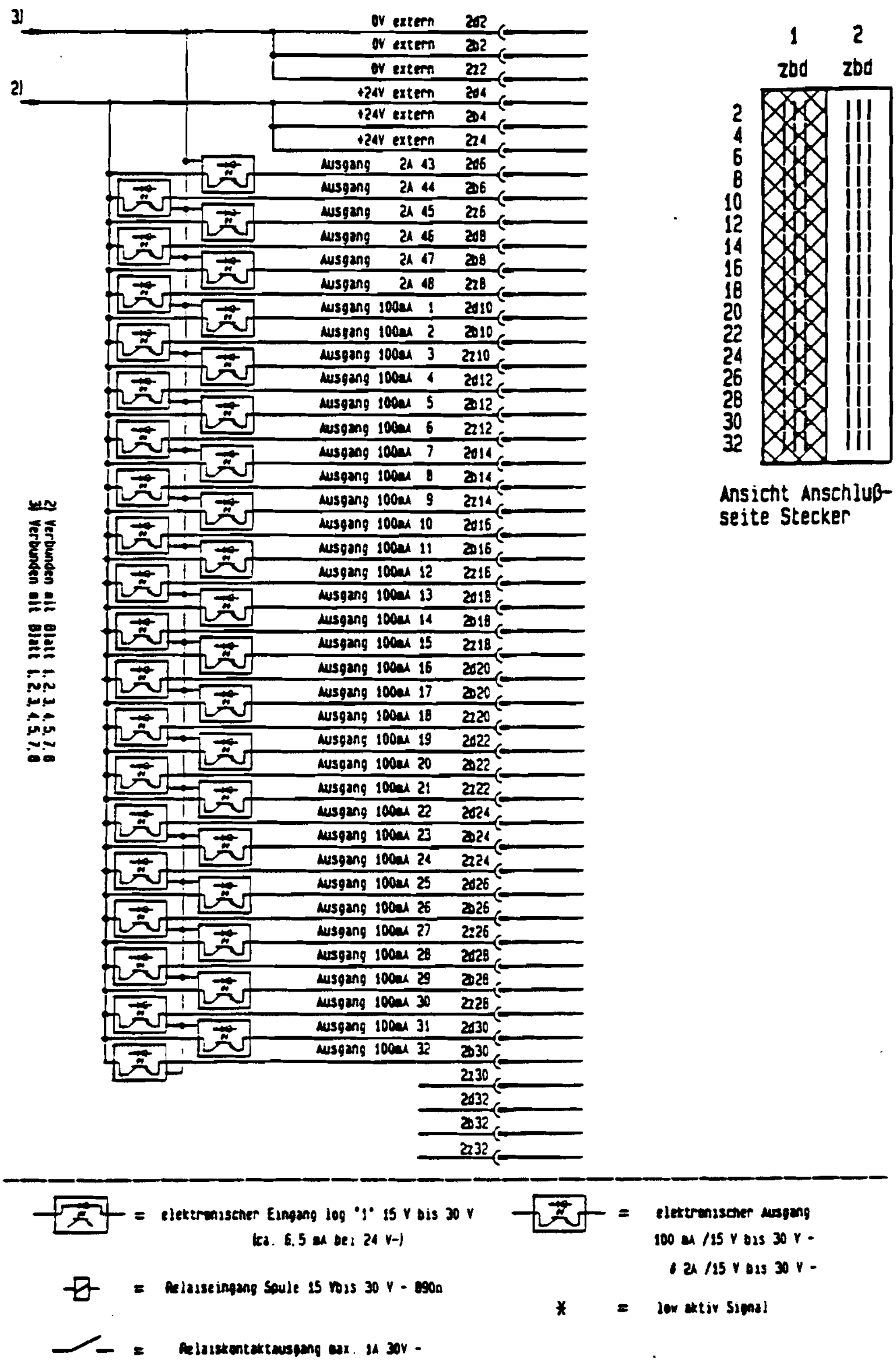
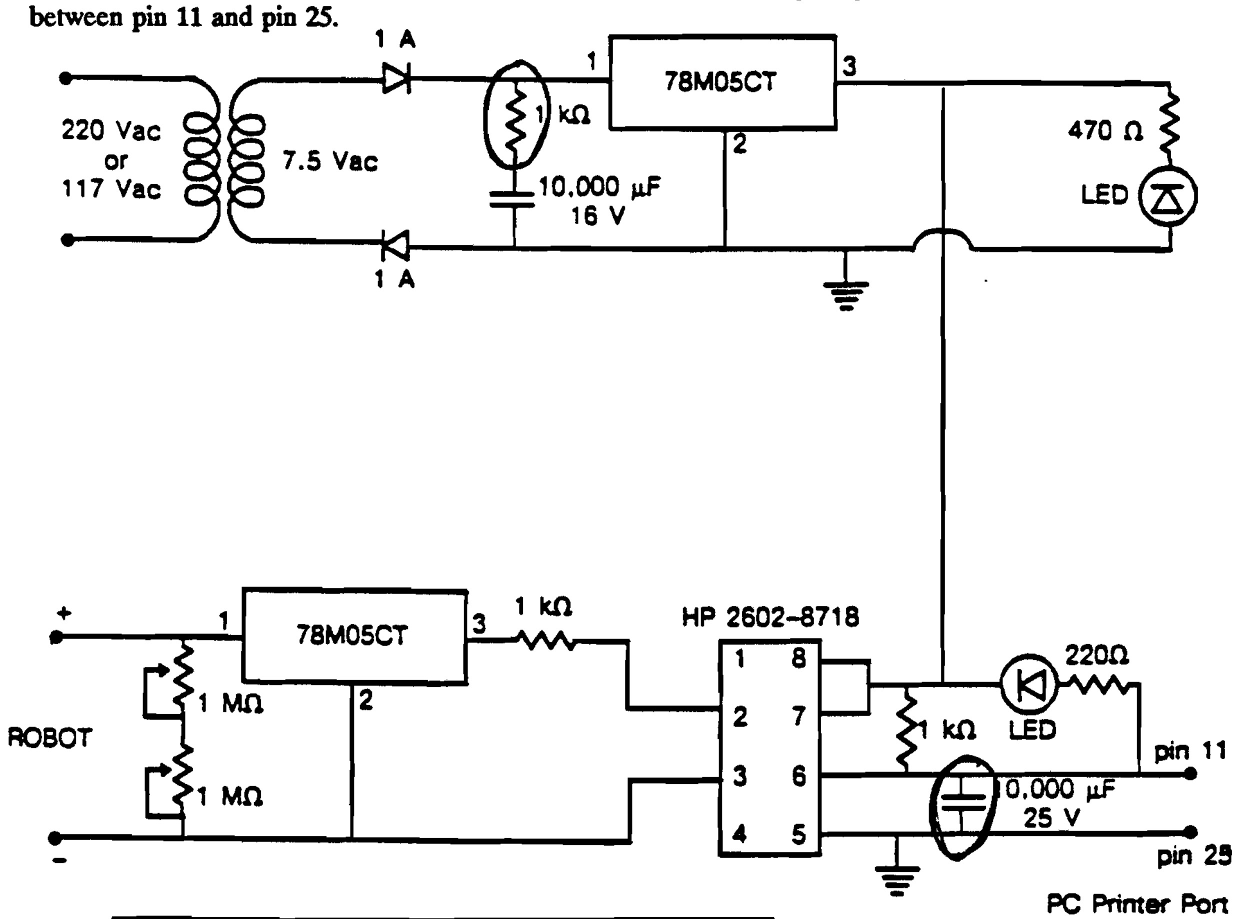


Figure A11.1. Robot Controller connection X13.

Appendix 12. ROBOT/PC interface schematic.

The schematic of the ROBOT/PC interface box. The interface change the Robot Controller I/O signal (between 15 and 30 V, probably 24 V) into an interrupt that can be received by the PC. The interface has an additional fuse on the input current. Only the schematic was included in the user guide, the interface had to be build. This schematic is the original McDonnell Douglas version. To make it work, I had to remove the encircled items: the resistor in front of the voltage regulator and the capacitor between pin 11 and pin 25.



Parts List

1	220 V - 7.5 V or 117 V - 7.5 V Transformer
2	1 A Diodes
3	1 KΩ Resisters
1	220 Ω Resisters
1	470 Ω Resisters
2	1 MΩ Potentiometers
1	10,000 µF 16 V Capacitor
1	10,000 µF 25 V Capacitor
2	5 mm LEDs (red and green)
1	HP 2602-8718 Optocoder
2	78M05CT Voltage Regulators (5 V, 500 mA)
1	Connection to robot controller
1	Connection to PC (RS-232 plug / socket)

Figure A12.1. Schematic of the ROBOT/PC interface box.

Appendix 13. The TIM-file.

The TIM-file is generated by the CTA-PC module and is called KUKA.TIM.

```
0.000  100.000   6  10   10.000  PCNT  JOINTSM
0.000  100.000   10.000  M/MIN  KUKACART
 0.4930    338.9094    58.8908    11.9107
 0.1942    303.2808    98.2234    23.8397
 0.1719    241.9583   117.8742    35.8251
 0.1749    249.5066   131.1367    48.1346
 0.1717    236.3820   148.4692    59.5170
 0.1791    256.4912   155.0120    71.7878
 0.1787    251.4872   163.9872    83.0113
 0.1696    242.5858   170.1315    94.7816
 0.1730    242.6495   181.0558   103.2985
 0.1624    233.2699   177.0049   120.6030
 0.0000     27.0738    48.5714     9.6207
 0.1619    150.0807    72.7815    19.2912
 0.2210    214.6317    87.8170    29.0292
 0.1667    158.0626    99.0683    38.8335
 0.1973    180.8275   108.9816    47.9494
 0.1791    169.6835   113.8750    57.7062
 0.1899    174.9723   118.3111    66.4728
 0.1732    161.5632   118.5800    77.6699
 0.1754    160.1404   123.6593    87.7605
 0.1721    159.8953   124.0682   104.0312
 0.3377   1613.3207    78.5094    14.7626
 0.2184    552.9589   121.1171    29.7202
 0.1979    426.7161   152.9034    44.5471
 0.1780    342.0509   173.1385    59.5547
 0.1808    326.6325   187.3997    74.5681
 0.1867    350.9213   203.4747    89.2637
 0.1626    308.5967   216.1540   104.0526
 0.1644    306.9101   228.7865   117.0536
 0.1564    299.8480   234.1499   130.8112
 0.1709    317.0256   245.2443   144.5954
 0.2601    305.2716    64.0144    14.4857
 0.1344    168.8941   102.0510    29.0961
 0.1723    241.4610   132.3292    43.6559
 0.1964    298.2966   156.1046    58.2490
 0.1902    287.1251   178.0605    72.7630
 0.2155    324.4430   192.1660    87.3017
 0.2140    329.1176   204.0398   102.4829
 0.2171    335.7048   222.2580   115.4532
 0.2194    340.1445   224.6018   132.1807
 0.2277    354.5379   237.7392   144.0448
 0.4008   5951.9175    59.2741    14.0913
 0.1360    162.9586    98.3732    28.2991
 0.1801    249.0397   127.1524    42.5324
 0.2363    340.8550   147.7030    56.9287
 0.2066    304.5757   170.4697    70.9537
 0.2276    335.6162   187.5414    84.8870
 0.2171    321.9691   198.5465   100.0598
 0.2416    359.0689   210.4418   113.2248
```

0.2355	354.7448	216.7926	129.7949
0.2477	372.0510	231.2254	141.6039
0.1519	179.0765	75.2075	17.8946
0.2196	376.7237	125.0274	35.8609
0.2174	374.3593	162.7487	53.9299
0.2309	419.7095	191.8844	71.8617
0.2426	456.9751	214.8175	90.0723
0.2302	423.1652	241.4964	107.1702
0.2206	410.0351	253.0565	125.3008
0.2271	408.8598	266.1037	144.0406
0.2349	444.7992	289.8425	158.3402
0.2372	439.0294	293.7101	178.8294
0.4609	1677.7722	349.2196	166.5520
0.2721	774.5982	469.9942	333.9252
0.2629	755.2845	537.8602	501.2703
0.2577	742.9110	582.5465	658.9147
0.2464	710.8800	580.1643	910.3048
0.2509	732.0033	625.0907	916.0662
0.2510	715.7963	641.3167	1204.3011
0.2505	654.6669	582.7220	1131.3131
0.2508	732.3884	620.0493	1131.3131
0.2319	603.4160	539.0150	1076.9231

Appendix 14. Correspondence.

The three letters that were send to the McDonnell Douglas Information Systems in Paris for help.

Mr. Eric Nicole.
McDonnel Douglas Information Systems.
106, Bureaux de la Colinne.
92213 St-Cloud.
France.

Eindhoven, 17-07-1991.

Dear Mister Nicole,

My name is Tijs Willems and i am one of the two students that are currently working with Robotics. It is my assignment to run CTA on the Kuka robot of the Falc-cell of the TUE which has been modelled by Hein-Jan van Veldhoven.

The manual timing file generation method worked very good but a problem occured trying to run the empirical data collection method. After typing the name of the options file that is used cta prompts: "error reading options files".

We tried several configurations of the options file:
- All uppercase (capital) letters like in the example in the manual. We even tried it one time with a ; at the end of every line.
- All lowercase letters (with and without ;).
- A mixed one with lowercase letters for the names of the files (because the filenames in the directory are also lowercase letters). This one is included in this letter.

All the files reffered to in the options file are in the userdirectory that is used. This directory contains all the files needed for modelling a cell.

We include the last options file we tried. This one is simular to the options template file that is in the systemlibrary. Could you please check this options file for errors and tell us how to solve this problem.

With kind regards

Tijs Willems

Mr. Eric Nicole.
McDonnel Douglas Information Systems.
106, Bureaux de la Colinne.
92213 St-Cloud.
France.

Eindhoven, 25-07-1991.

Dear Eric,

I used the options template file to make a new options file. This time i used a conversionfactor of 1.0 as you suggested and this time it worked. I got curious and tried several other things and they all worked! I even tried to change the conversionfactor and still it didn't go wrong. I still wonder what the mistake(s) i made was?

Now another problem occured. All the sequences that cta made are empty. If for examle the first data line of the options-file is :

```
   DATA = 30 60 50 0 0 0 0 20 1.0 20 10.0
```

the sequence for joint 1 is:

```
   GOTO_JOINTS: (IN),0 ,0 ,0 ,0 ,0 ,0 ,NOP;
   GOTO_JOINTS: (IN),0 ,0 ,0 ,0 ,0 ,0 ,OUTLAW:
    + 39 times this last line
```

The number of goto joints are correct but all angles are zero. I can't find out why this happens.

As for the robot/pc interface, I allready made it. In the CTA-manual release 7.0 i found an electronic schematic. This i used to build it. I also found out that that schematic is not fully correct. I took out two things:

- the resistor between the transformer and the voltage-regulator.
- the capacitor at the printer-port.

Now it works very good!

The subject of my study is indeed just evaluation of CTA. That is, run CTA on the falc-cell modelled by H.J. van Veldhoven. But it is not my graduation-study. It's a small study before a graduation-study.

Thanks for the quick response and all the information you sent me. I hope you can help this time too ?

Best regards,

Tijs Willems.

Mr. Eric Nicole.
McDonnel Douglas Information Systems.
106, Bureaux de la Colinne.
92213 St-cloud.
France.

Eindhoven, 09-09-1991.

Dear Eric,

We have run in to two problems this time. The first one is a CTA-one. The last data-line in the optionsfile is for straight-line movement. The data represents the position and the rotation of the reference tpoint for straight-line movement. Every time i give the reference tpoint a certain position and rotation (for example 0 0 0 0 0 0) CTA makes a sequence and a cell where the position of the reference tpoint is good (0 0 0) but the rotation is not the same. The tpoint is rotated about the Y-axis by 90 degrees. This is not happening if you create a tpoint with the option 'position' in PLACE. I can't find out why this happens.

The second problem is a tranlation problem. When we write a USR-file containing commands like pause, delay, speed, etc. the translation to SRCL-language fails because of translation errors on those commands. We don't know what the syntax must be for commands that have no & or $ in front of the command. Perhaps we need the operational description manual for our translator.

I hope you can help us and send us the operation description manual.

Best regards,

Tijs Willems.