HIGHLY ACCURATE INTEGRATION OF TRACK MOTIONS

Michael Kleinkes, Angela Lilienthal, Werner Neddermeyer
University of Applied Science in Gelsenkirchen
Neidenburger Str. 43, 45897 Gelsenkirchen

Keywords: Industrial Robots, 7^{th} axis, linear track, offline programming.

Abstract: According to the largeness of the workpieces in several industrial environments, a great number of industrial robots is placed on external track motions, so called 7^{th} axes, as for automotive or aircraft industries. Flexible automation today requires absolute high accuracy. For example modern robotics deals with offline programming, copying programs between similar working cells, reflecting programs or image processing for 3D-pose estimation. All these tasks need absolute high accuracy and in fact, there have been many investigations for increasing the accuracy of single robots in the past few years.
In contrast to that the use of track motions will dramatically increase the position error and badly influence the static behaviour of the robot system. The main reasons for these additional errors are the incorrect identification of the main track direction and furthermore, very crucial, the non-linearities of the TCP (Tool Center Point) during the robots motion on the track. This article will introduce a new method of identification and mathematical integration of linear tracks. At first we present the method for measuring and generating profiles of single tracks by making use of the discrete fourier transformation (DFT) and cubic spline interpolation. Then a method for recalculating offline generated programs for real environments is presented, followed by a method for copying programs taking two profiles of track motions into consideration. Finally some measurement results are shown.

1 INTRODUCTION

Increasing demands on the flexible automation, e.g. offline programming of robot motions or the application of more and more complex tasks like sealing of carbodies, require robot systems with absolute high accuracies. In the past there was done a lot of research work and investigations to increase the accuracy of robots by an individual identification (Denavit and Hartenberg, 1955), (Maas, 1997) of internal parameters, so that robots with a high accuracy option are available (Nitschke, 2002). From the experience the inaccuracy of these robots is almost less than 1 mm.

In many cases of the automotive industry, the robot's workspace is extended by external track motions which deteriorates the accuracy of the system in a significant manner about several millimetres or more. This alarming fact is not compatible to the demands on the flexible automation at all and additionally is neglected by scientists and robot manufactorers as well.

Offline programming, copying or reflecting programs or 3D pose estimation requires a coordination of the track and the robot motion, that means an integration into the robots coordinate system. The methods of measurement, identification and mathematical integration of track motions are shown exemplary for an ABB robot. Similar computations for other types of robots, e.g. KUKA or FANUC, are possible as well. Figure 1 will give you an overview of the important robot coordinate systems.

The alignment frame, as shown in figure 1, provides the connection of the track motion to the robot. The ABB controller is computing the main track direction from a frame containing quaternions (Convay and Smith, 2003), which indicates the rotation from the robot's world frame to the track. It is obvious, that the similar parameters for the track integration have to be worked out for each type of robot, e.g. KUKA, COMAU or FANUC. It is an essential knowledge, that any type of controller unit is modelling tracks as straight lines, where the incorrect identification of the main track direction and - very crucial - the non-linearities of the robot's TCP during a motion on the track, the so called robot's twist, are not

Kleinkes M., Lilienthal A. and Neddermeyer W. (2005).
HIGHLY ACCURATE INTEGRATION OF TRACK MOTIONS.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 11-15
DOI: 10.5220/0001169100110015

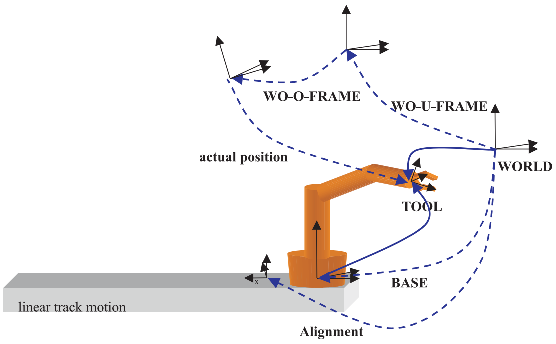

Figure 1: Coordinate systems

taken into consideration. The offline generated robot motions, the actual positions, are related to the workpiece, which is provided by the so called workobject (WO-U- and -O-FRAME) of the robot controller interface.

2 IDENTIFICATION OF TRACK MOTIONS

For the identification of track motions we make use of a mobile coordinate measurement device, e.g. a laser tracker. Due to the computation of the robot's behaviour under motion on the track, it is aimed to identify the robot's coordinate system at n positions. For that, first the optimal number of positions is to be found by the spectral analysis (Press et al., 1989) of a fine scanned motion.

2.1 Fine scanning

For the evaluation of the optimal number of positions where a frame identification is necessary, a single high resolution scan of the robots TCP during a motion on a straight line with constant joint configuration, that means movement only by the track, is generated. The movement is carried out with low and constant velocity V and the scan mode is switched on high resolution and discrete in time (sampling interval: dt).

Assuming that one axis of the measurement frame points in the direction of the expected line, the sum of the 3D coordinates x_k, y_k and z_k contains the command function - a ramp function s_k - plus the non linear twist motion, which is of interest. After the elimination of the ramp function, the Fourier transform of the h_k's contains the spectrum of the non-linearities of the track:

From

$$s_k = k \cdot \Delta, k = 0, 1, ..., N-1 \quad (1)$$

and

$$\Delta = V \cdot dt \quad (2)$$

follows

$$h_k = x_k + y_k + z_k - s_k \quad (3)$$

where the x_k, y_k and z_k are the 3D scan of the TCP (see figure 2).

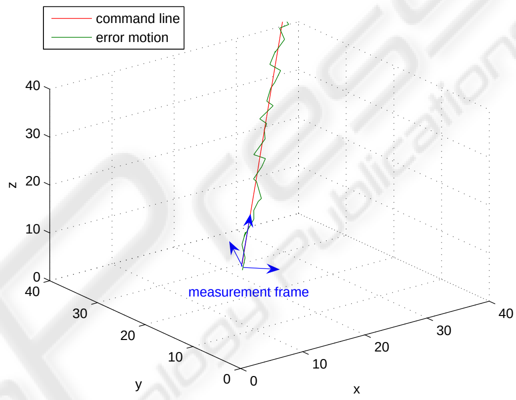

Figure 2: command and error motion

The DFT becomes

$$H(f_n) = X(f_n) + Y(f_n) + Z(f_n) - S(f_n) \quad (4)$$

from

$$f_n = \frac{n}{N\Delta}; n = -\frac{N}{2}, ..., \frac{N}{2} \quad (5)$$

follows that

$$\begin{aligned} H(f_n) &= \int_{-\infty}^{\infty} h(s) e^{-j2\pi f_n s} ds \\ &\approx \sum_{k=0}^{N-1} h_k e^{-j2\pi f_n s_k} \Delta \\ &= \Delta \sum_{k=0}^{N-1} h_k e^{-j2\pi kn/N} \end{aligned} \quad (6)$$

From eq.5 it is obvious that the frequency is of the unit mm^{-1}.

Figure 3 gives an example about a spectrum taken from a typical arrangement.

If a detection of errors greater than 0.5 mm is required, it is obvious by the sampling theorem that it is essential to identify the robot coordinate frame each 100 mm ($= 1/(2 \cdot 0.005 mm^{-1})$).

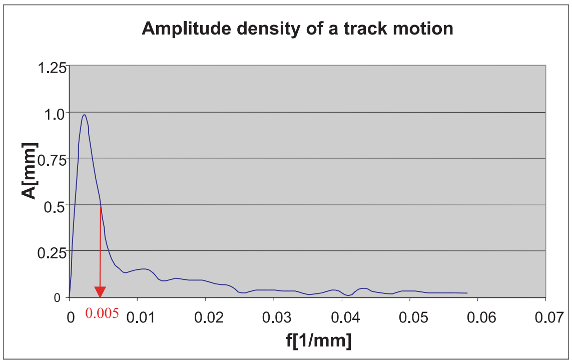

Figure 3: Example of a track spectrum

2.2 Generation of track profiles

Once the sampling rate is known, it is possible to generate the continuous of the non-linear robot behaviour. At first the robot's coordinate frame is identified at n positions on the track. The measurements are done using the same set up points at each position. That means the robot is taken as a rigid body, because of it's high repeatability.

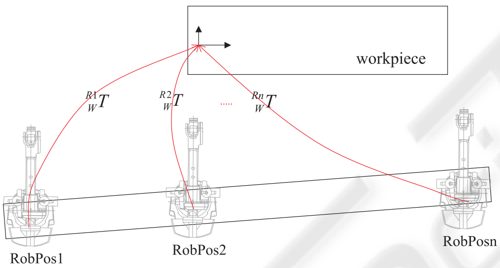

Figure 4: Measurement of the axis profile

Figure 4 shows the relations if the workpiece, to which the offline generated programs are related to, is already identified. The results are n transformations where the six dimensions of freedom are implicitly dependent on the value of the track:

$$ {}^{Ri}_{W}T = f_i(\alpha(l_i), \beta(l_i), \gamma(l_i), t_x(l_i), t_y(l_i), t_z(l_i)) \quad (7) $$

For the computation of a continuous description, we make use of the cubic spline interpolation procedure (Dautray, 2000), which connects the discrete pairs of varieties (α_i, l_i), (β_i, l_i), etc. to smooth and also smooth in the first derivative functions, so that the continuous form becomes:

$$ {}^{R}_{W}T(l) = f(\alpha(l), \beta(l), \gamma(l), t_x(l), t_y(l), t_z(l)) \quad (8) $$

Note, that these are the relations where the track is not coordinated to the robot motion. For a connection to the robot base, the axis transformation, which is dependent on the value l of the robot's position on the track as well, is computed by:

$$ T_{ax}(l) = \begin{pmatrix} I & l \cdot \vec{A}x_{Dir} \\ 0^T & 1 \end{pmatrix} \quad (9) $$

where I is the 3×3 identity matrix and $\vec{A}x_{Dir}$ is the controller owned normalised axis direction (which has to be worked out individually belonging to the type of robot). From eq.8 and 9 the workobject frame $WO(l)$ becomes:

$$ WO(l) = T_{ax}(l) \cdot {}^{R}_{W}T(l) \quad (10) $$

which is shown in figure 5.

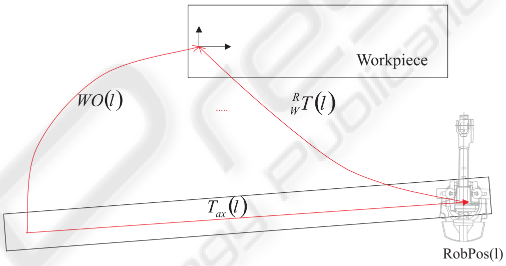

Figure 5: Connection to track motion

3 MODIFICATION OF ROBOT PROGRAMS

The modification of robot programs has to be divided into two parts. The first case is copying offline generated motions to a real profile, whereas the second is copying programs between similar working cells taking both - the master and the slave - profile into consideration.

3.1 Modulation of offline generated robot programs

In figure 6 you can see the robot's behaviour under a motion on it's 7^{th} axis. Note that the predefined controller owned workobject is WO1 which is valid for the robot position RobPos1 without moving on the track. But under robot motion and it's twist it points from the position RobPos n to the dashed workpiece which of course is not there.

From the splined workobject and the actual value on the track we know all relations, so as a consequence the corrected workframe becomes:

$$ T_{corr} = (WO1)^{-1} \cdot WO(l) \cdot T_{offline} \quad (11) $$

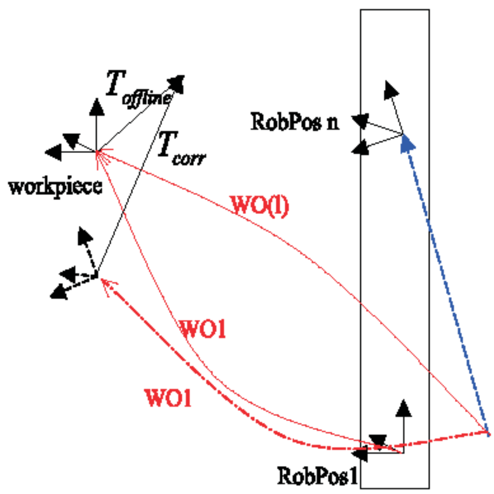

Figure 6: Workobjects and twist

The example shown in figure 7 illustrates the modifications to the motion commands in the robot language RAPID, which is characteristic for ABB robots.

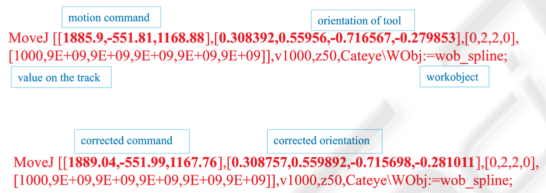

Figure 7: Modification of robot commands

The motion commands are always related to the coordinate system of the workpiece and the orientation of the tool is - a speciality of ABB robots - expressed in quaternion notation. The value on the track has to be given explicitly for each robot motion, because of the overdetermination of the inverse problem when using a 7^{th} axis - that means for sure an additional degree of freedom. The choice of the predefined workobject is individually possible for any command.

So- what the proposed procedure does is just to replace the motion commands - translation and orientation - with the corrected ones. That means that the corresponding software tools won't change the structure of the robot file. It will be a kind of artificial post teaching of robot programs.

3.2 Copying programs between similar robot systems

From chapter 3.1 we are able to modify offline generated robot files with the behaviour of the real robot and due to that, create accuracies for the robot systems, which are rather identical to those of single robots.

But what is to be done if two track motions are of interest in the case of copying programs from one station to another? The solution is the computation of so called *quasi offline* programs under consideration of the track from which the programs should be copied. That means: compute the inverse of the described method as illustrated in figure 8.

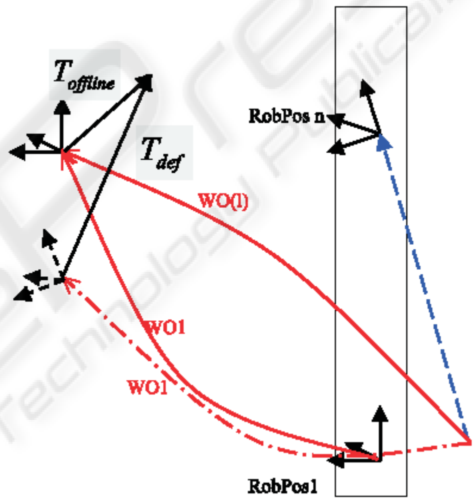

Figure 8: Copy programs

Now the defined or teached workframes T_{def} of the robot master are known which have to be recalculated into an ideal perpendicular offline world ($T_{offline}$):

$$T_{offline} = (WO(l))^{-1} \cdot WO1 \cdot T_{def} \qquad (12)$$

These quasi offline generated programs from the robot master profile can be taken as the input files for the method described in chapter 3.1 to create the converted motions for the slave system.

4 RESULTS

Figure 9 illustrates the results of a test program containing 12 work frames. The blue columns show the errors of the conventional robot system whereas the red ones show the errors of the modified workframes.

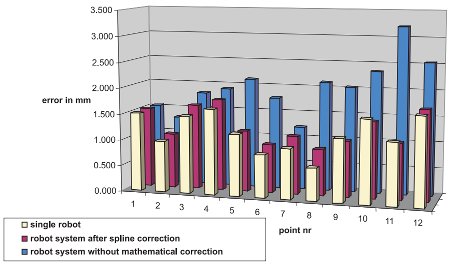

Figure 9: Results 1

For a validation the yellow ones show the errors of the robot itself.

So the final improvement is demonstrated by figure 10. The errors caused only by the track are shown by the bright blue columns for the conventional system and - the dark blue ones - for the modified robot motions.

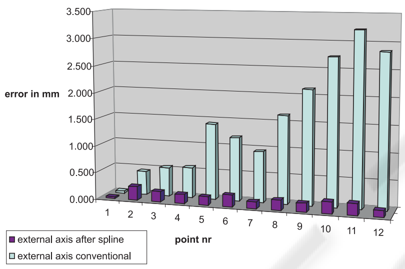

Figure 10: Results 2

5 CONCLUSION

As a conclusion we want to emphasise the compensation of errors caused by external track motions. That means the

- accuracy of the robot system is nearly identical to that of the single robot.
- the worst case for the inaccuracy of copied programs is the doubled inaccuracy of one single robot.

A future prospect will be the implementation into the robot control, that means including the measured splines directly into the position control of the robot as an option for high accurate robots. A further step will be the implementation into continuous path motions.

REFERENCES

Convay, J. and Smith, D. (2003). *On quaternions and octonions*. A K Peters, Massachusetts.

Dautray, R. (2000). *Mathematical analysis and numerical methods for science and technology*. Springer, London.

Denavit, J. and Hartenberg, R. (1955). A kinematic notation for lower-pair mechanism based on matrices. In *ASME Journal of Applied Mechanics 22*, pages 215–221.

Maas, H. G. (27.7.-1.8.1997). Dynamic photogrammetric calibration of industrial robots, spie's 42 annual meeting, san diego. In *Videometrics V, SPIE proceedings Series Vol. 3174*.

Nitschke, H. (2002). *Zur Bestimmung geometrischer Parameter von Industrierobotern*. Bayrische Akademie der Wissenschaften, Dissertation, TU Munich.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. F. (1989). *Numerical Recipes in C*. Cambridge University Press.