Optimized Control Methods for Capturing Flying Objects with a Cartesian Robot

Heinz Frank, Dennis Barteit, Marcus Meyer,
Anton Mittnacht
Electrical Engineering
Reinhold-Würth-University
Künzelsau, Germany

Gregor Novak, Stefan Mahlknecht
Institute of Computer Technology
Vienna University of Technology
Vienna, Austria

***Abstract*— Throwing or shooting is a new approach for the transportation of objects within production systems. Since Gantry Robots are often applied to load and unload machines, in a research project a Cartesian Robot was used for capturing flying objects. A camera system is measuring the object's positions during a throw in subsequent periods of time. Based on these measurements it can predict the capturing point with an increasing accuracy. So it can direct the robot during a throw to the capturing point also with an increasing accuracy. In this paper control methods are proposed which allow in such an application fast and smooth motions of the robot to the final capturing point. For the evaluation of the control methods two scenarios, a simplified and a realistic one, are defined.**

***Keywords*—Cartesian robot, flying objects, capturing, control method**

I. INTRODUCTION

Throwing or shooting is a new approach for the transportation of objects within production systems [1]. Fig. 1 shows our concept for this method. For this we have already presented in [2] a throwing device and two different types of capturing devices. For the visual tracking of flying objects during a throw, different methods are already described in several other scientific works [3, 4, 5]. The focus in this paper is laid on the optimized control of the robot for tracking the capturing device to the final capturing point.

The features of the concept presented in Fig. 1 can be summarized as follows:

- Objects are thrown with different shapes and masses up to 60 g over distances of about 3 m. Since the objects shall be transported very fast, they shall be thrown with speeds up to 10 m/s. With that very flat trajectories can be achieved. The flight-time of the objects on such trajectories is about 300 ms.
- When throwing objects, which are unsymmetrical and not absolute identical, their trajectories are depending on sensitive influences like different conditions during the acceleration by the throwing device, the influence of the gravitation and the aerodynamic resistance.

Fig. 2 shows as an example two trajectories of the same electrical terminal block. After 3 m they have in the z-axis a deviation of 120 mm. For other objects it shall be assumed, that they arrive at the robot in a capturing area which is 400 mm x 400 mm.

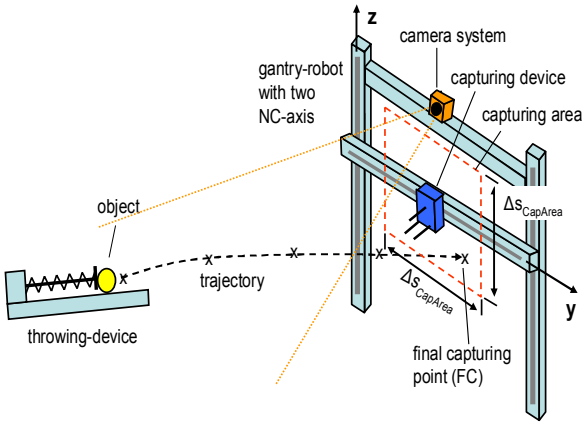

Fig. 1. Concept for Throwing Objects

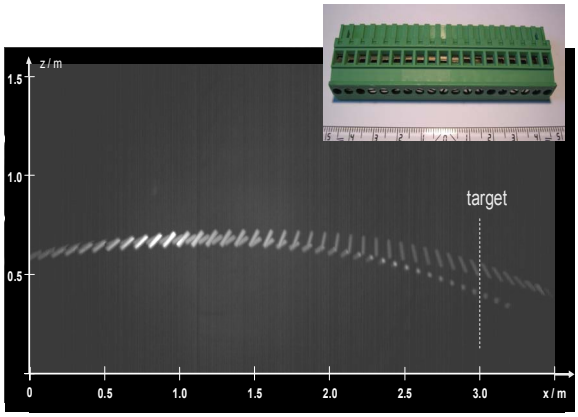

Fig. 2. Two Trajectories of an electrical terminal block

- In production systems gantry-robots are often applied to load and unload machines. Therefore a Cartesian robot is used to capture the flying objects. The robot in Fig. 3 has the following technical data: The working-area in the y-axis is 1000 mm and in the z-axis 800 mm. The maximum acceleration and the maximum speed of this axis are $a_{max} = 25$ m/s^2 and $v_{max} = 4$ m/s at a payload of m = 5 kg.

In this paper control algorithms are presented, which allow for this application fast and smooth movements of the Cartesian robot.

II. Related Work

The visual tracking of flying objects and the tracking of NC-axis of robots have already been realized in several previous research works. In [3, 4, 5, 6, 7, 8] studies are described, where balls or arrows are captured or batted. A paper, where the transportation of objects within productions systems by throwing and capturing was proposed, is not known.

To get very short transportation times respectively flight times, the flight-speed was specified as $v_{max} = 10$ m/s. This is a high requirement which is met up to now only by few existing systems. One of them is described by Taku in [6]. In this project a robot is able to bat flying Styrofoam-balls with speeds up to v = 8 m/s over distances of 2,5 m. In the opposite of our Cartesian robot a 5 DOF robot, consisting of revolution and bending motion alternately, is used. An application were flying objects are captured by a Cartesian Robot is not known.

According to [6] a main requirement for high-speed motions of robots is to achieve with the features of the real robot not only fast but also smooth motions. For the transportation of objects in production systems with short cycle times, smooth motions are extra important to preserve the mechanical components of the robot for a long life time [9].

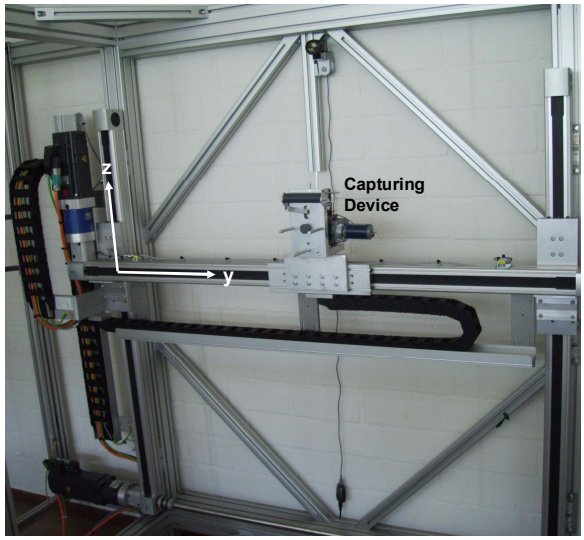

Fig. 3. Cartesian robot for capturing flying objects

Only few papers deal with optimized algorithms for the control of the robots in such applications. The control-method described in [6] is related to robots with rotation axis. In [7] for a humanoid robot a bell-shaped velocity-profile is presented. With this proposal "human-like" (smooth) motions can be achieved, however it is not optimized to fast speeds.

III. Basic Requirements To The Cartesian Robot

In the approach as it is shown in Fig. 1, at the beginning the robot is positioned in the center point of the capturing area. After the point of time, when the object is thrown, the robot must be capable to move the capturing device during the flight time Δt_{fl} to each point within the capturing area. This means that each of the two numerical controlled axis of the Cartesian robot (NC-axis y and z) must be able to move within Δt_{fl} a distance of $\Delta s_{axis} = ½ * \Delta s_{CapArea}$ (Fig. 1). Since these distances are short, the accelerations of the NC-axis are more important for that, than their maximum speeds. Fig. 4 shows for a NC-axis the required times Δt_{axis} over the distances Δs_{axis} to be moved at different accelerations. For this it was assumed, that the axis are accelerated with a constant value a_i during the first half of the distance and that they are decelerated with the same value during the second half of the distance. The maximum speeds, which are achieved during the movements, are also shown in Fig. 4. Fig. 5 shows the s-curves for such a movement, when the y-axis of the Cartesian robot moves with an acceleration a = 25 m/s^2 over a distance of $\Delta s_{axis} = 200$ mm.

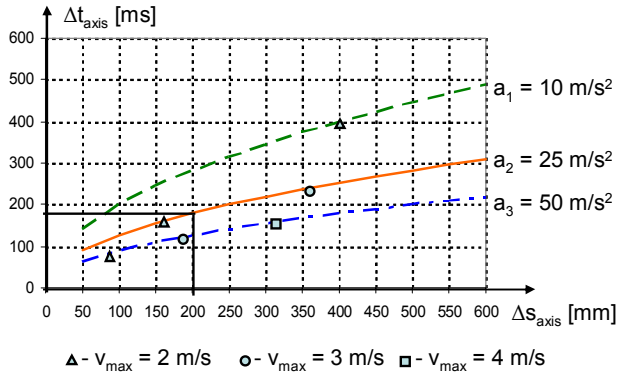

Fig. 4. Positioning times of a linear NC-axis for different accelerations a_i and unlimited speeds v

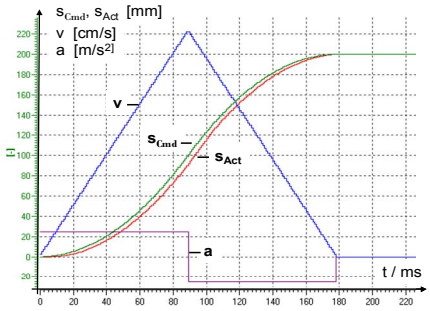

Fig. 5. S-curves of the y-axis of the Cartesian robot over s = 200 mm with a = 25 m/s^2 (s_{Cmd} – command position; s_{Act} – actual position)

IV. Prediction Of The Capturing Points

A. General Sequence of the Predictions

During the flight of an object, the positions on its trajectory are measured by a camera-system in equidistant periods of time $\Delta t_{Predict}$. With the information of these subsequent measurements the camera-system predicts the capturing-point for the robot. At the beginning of such a measurement-series the predictions are still inaccurate. With the increasing number of measurements the predictions are becoming more and more accurate. This is illustrated in Fig. 6 with a simplified example. At the beginning, the capturing device is at the center point of the capturing area (CP). It is assumed that the camera system makes subsequently four predictions for the capturing point (PC_{ti}). At the first prediction (PC_{t1}), the accuracy is still low. Therefore the area of prediction (AoP_1) is still big. The area of prediction is defined as an area, in which the final capturing point will be expected with a certain statistical probability. With the consecutive predictions at t_2, t_3 and t_4 the accuracy of the predicted capturing points becomes higher and so the areas of prediction become smaller. In Fig. 6 the final capturing point (FC) is assumed in the corner of the capturing area, which is the worst possible position for the robot.

To avoid in the following descriptions a double consideration of the y- and the z-axis, it shall be assumed that the characteristics of the flying objects and the characteristics of the robot are in the y-direction and in the z-direction identical. So only a general s-axis has to be considered.

B. Definition of Fix Scenarios

During the flight the information which the robot receives from the camera system can be described by the following features:

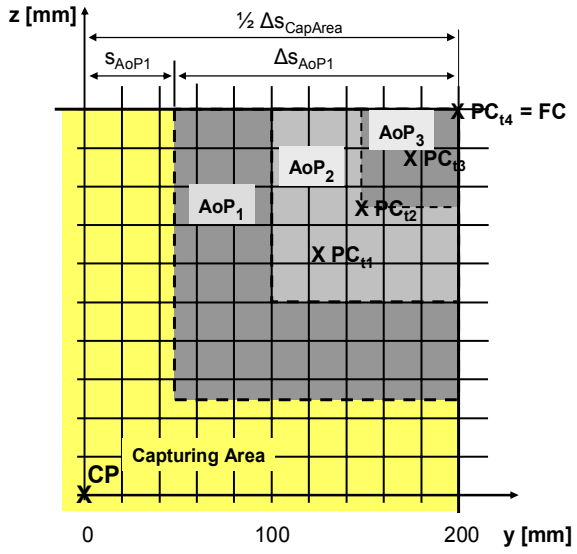

Fig. 6. Simplified example for the prediction of a capturing point by a camera system.
CP - Center Point of the Capturing Area;
PC_{ti} - Predicted Capturing Point at time ti;
AoP - Area of Prediction;
FC - Final Capturing Point

- Period of time in which the robot receives new values ($\Delta t_{Predict}$),
- increase of the accuracy of the subsequently predicted capturing points and
- irregularity of the positions of the subsequently predicted capturing points.

For the comparison of different control methods fix sequences of information from the camera system to the robot shall be defined. They shall be used to evaluate each method with the same input data.

For the motion of the robot always its worst case from the center point (CP) to a final capturing point (FP) in the corner of the capturing area, shall be considered (Fig. 6).

For the increase of the accuracy of the subsequently predicted capturing points two cases shall be distinguished:

- Case 1: Linear increase of the prediction accuracy (simplified situation)

  It shall be assumed that the accuracy of the predicted capturing points increases linear. The side-length of the areas of prediction (Δs_{AoP}) shall become linear smaller (see example in Fig. 6). After the duration of a time Δt_{FC} the error of the predicted capturing point shall be zero.

$$\Delta s_{AoP} = 0{,}5 \cdot \Delta s_{CapArea} - \frac{0{,}5 \cdot \Delta s_{CapArea}}{\Delta t_{FC}} \cdot t \qquad (1)$$

- Case 2: Exponential increase of the prediction accuracy (realistic situation)

  Our experiments have shown that the accuracy of the predicted capturing points increases with an exponential function. For that the side-length of the areas of prediction can be calculated as follows:

$$\Delta s_{AoP} = \frac{0{,}5 \cdot \Delta s_{CapArea}}{\Delta t_{FC}^{\,2}} \cdot (t - \Delta t_{FC})^2 \qquad (2)$$

For the irregularity of the positions of the predicted capturing points we also want to distinguish two different cases:

- Case A: Regular approximation to the final capturing point (simplified situation)

  It shall be assumed that the subsequently predicted capturing-points have a regular approximation to the corner of the capturing area (see example in Fig. 6). Therefore the positions of the predicted capturing points shall be calculated as follows:

$$s_{PC,i} = 0{,}5 \cdot \Delta s_{CapArea} - 0{,}5 \cdot \Delta s_{AoP,i} \qquad (3)$$

- Case B: Irregular approximation to the final capturing point (realistic situation)

  It shall be assumed that the subsequently predicted capturing points have an irregular approximation to the corner of the capturing area. Therefore the following equations shall be used:

  If i is odd:

$$s_{PC,i} = 0{,}5 \cdot \Delta s_{CapArea} - 0{,}5 \cdot \Delta s_{AoP,i} \quad (4)$$

  If i is even:

$$s_{PC,i} = 0{,}5 \cdot \Delta s_{CapArea} - 0{,}5 \cdot \Delta s_{AoP,i} + d \cdot (\Delta s_{AoP,i} - \Delta s_{AoP,i+1}) \quad (5)$$

For the evaluation of the control methods finally only the following two combinations shall be applied:

- Scenario 1A: Simplified sequence (Fig. 7)

  The simplest scenario is the combination of the linear increase of the prediction accuracy (case 1) with a regular approximation of the predicted capturing points to the final capturing point (case A). This scenario shall be used only to explain the control methods. In this scenario $\Delta t_{Predict}$ shall be assumed as 50 ms.

- Scenario 2B: Realistic sequence (Fig. 7)

  For a realistic scenario the exponential increase of the prediction accuracy (case 2) shall be combined with an irregular approximation of the predicted capturing points to the final capturing point (case B). In this scenario $\Delta t_{Predict}$ shall be assumed as 25 ms, as it is in the real system.

V. Control System

For the control of the Cartesian robot a commercial available motion controller SIMOTION P350 is used (Fig. 8). It runs on a PC with a 2 GHz Pentium processor. The s-curves of the command positions for the movements (s_{Cmd}) are generated by an interpolator for each NC-axis separately. The scan time for the position control is 1 ms. The speed control is implemented in the drive unit and has a scan time of 125 µs.

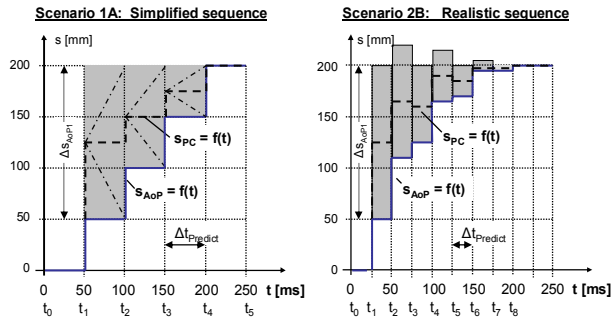

Fig. 7. Scenarios for the evaluation of control methods
s_{PC} – position of the predicted capturing point
s_{AoP} – position of the boundary of the area of prediction

In the controller to each period of time $\Delta t_{Predict}$, in which the robot receives from the camera system a new prediction for the capturing point, a new "section of movement" (SoM) for the robot is allocated. At the beginning of such a SoM each NC-axis gets a new target-position and independent whether a NC-axis is just standing, accelerating, moving with a constant speed or decelerating, the interpolator will lead the NC-axis from its actual state on new s-curve directly to the new target position. Therefore it calculates for each axis the command positions s_{Cmd} completely new.

VI. Control Methods

A. *Predicted Capturing Points as Target Points*

The most simple control method is to use in each SoM the position of the new predicted capturing point (in Fig. 7: $s_{PC} = f(t)$) as the new target-position for the NC-axis. This causes very unsmooth movements as it can be explained with a simple example in Fig. 7, Scenario 1A. At time t_1 the target-position is $s_{PC} = 125$ mm. At time t_2 a new predicted capturing point must be expected between $s_{PC} = 50$ mm and $s_{PC} = 200$ mm. So with each new predicted capturing point a turn of the moving direction is possible. Therefore this method shall not be considered in more detail.

B. *Boundaries of the Areas of Prediction as Target Points*

To avoid turns in the moving directions from one SoM to the next, we propose to use the boundaries of the area of predictions as target-positions (in Fig. 7: $s_{AoP} = f(t)$). These boundaries may be used as target-positions only if the robot can reach them from outside the area of prediction. With this demand at the beginning of a new SoM a s-curve is generated which leads the NC-axis to the boundary of the actual area of prediction. If during this movement a new prediction is delivered from the camera system, a new s-curve is generated for the next SoM.

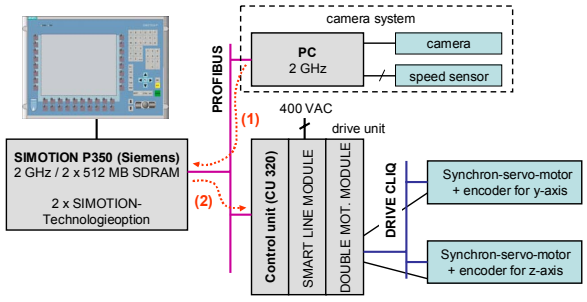

Fig. 8. Control system for the Cartesian robot. Every time, when the SIMOTION receives from the camera system a new prediction for the capturing-point (1), it starts a new SoM. For such a new SoM i it calculates according to one of the described control algorithms new values for $\Delta s_{Move,i}$, $v_{End,i}$ respectively a_i. Based on the actual state of the NC-axis and these values a new s-curve will be generated (2).

In Fig. 9 and Fig. 10 the results of this control method are shown for the scenarios 1A and 2B. Both figures are showing no simulations but real movements of the y-axis of the Cartesian robot. In the realistic scenario (Fig. 10) the robot can move the capturing device within 230 ms to the corner of the capturing area. This is a good result for capturing objects according to the specifications in section I. The disadvantage of this method however is, that in both scenarios there are a lot of acceleration- and deceleration-steps, which cause a high mechanical stress to the robot.

C. *Limitation of the Speeds*

With a specific limitation of the speed in each SoM it can be achieved, that one SoM can not have both an acceleration and also a deceleration step.

At the beginning of a SoM i the distance which a NC-axis can move (Δs_i^*) consists of a remaining distance from the previous SoM ($\Delta s_{Remain,i-1}$) and a new distance according to the previous and the new prediction values :

$$\Delta s_i^* = \Delta s_{Remain,i-1} + (s_{AoP,i} - s_{AoP,i-1}) \quad (6)$$

This distance Δs_i^* shall be divided in a distance $\Delta s_{Move,i}$ which shall be moved real during the SoM i and in an eventually needed distance for the deceleration, if $s_{AoP,i+1}$ will become equal to $s_{AoP,i}$:

$$\Delta s_i^* = \Delta s_{Move,i} + 0{,}5 \cdot \frac{v_{End,i}^2}{a_{max}} \quad (7)$$

With $\Delta s_{Move,i}$ the following cases must be distinguished:

- If $\Delta s_{Move,i} > v_{End,i-1} \cdot \Delta t_{Pred} + 0{,}5 \cdot a_{max} \cdot \Delta t_{Pred}^2$, the NC-axis can be accelerated with a_{max} during the whole SoM i.
- If $\Delta s_{Move,i} > v_{End,i-1} \cdot \Delta t_{Pred}$ and $\Delta s_{Move,i} < v_{End,i-1} \cdot \Delta t_{Pred} + 0{,}5 \cdot a_{max} \cdot \Delta t_{Pred}^2$, the acceleration in the SoM i shall be limited to the following speed:

$$v_{End,i} = \frac{\Delta s_{Move,i} + \frac{v_{End,i-1}^2}{2 \cdot a_{max}}}{\frac{v_{End,i-1}}{a_{max}} + \Delta t_{Pred}} \quad (8)$$

- If $\Delta s_{Move,i} = v_{End,i-1} \cdot \Delta t_{Pred}$, the speed during the SoM i shall be constant

$$v_{End,i} = v_{End,i-1} \quad (9)$$

- If $0 < \Delta s_{Move,i} < v_{End,i-1} \cdot \Delta t_{Pred}$, the speed shall be decelerated to

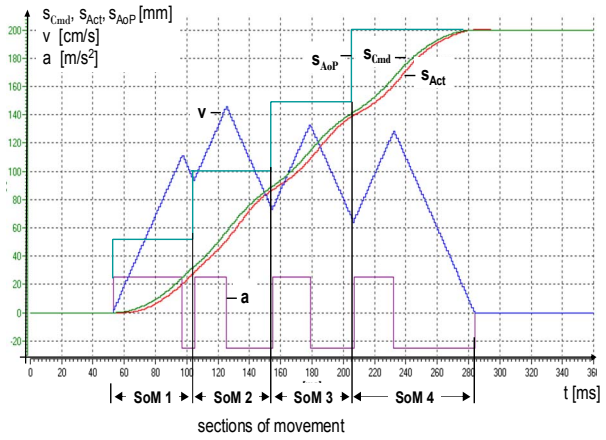

Fig. 9. Movement according scenario 1A with v_{max} and a_{max} (s_{Cmd} - command position; s_{Act} – actual position; v – speed; a – acceleration; s_{AoP} – positions of the boundary of the area of prediction)

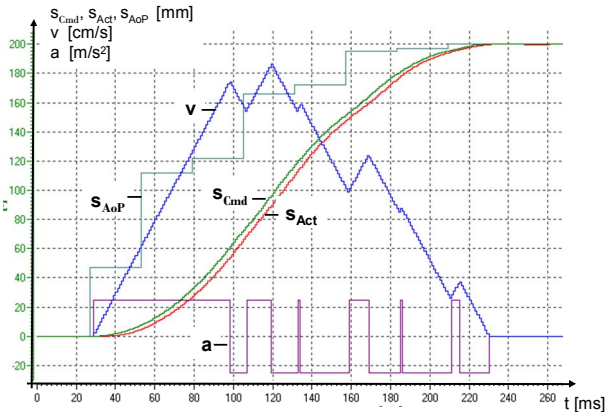

Fig. 10. Movement according scenario 2B with v_{max} and a_{max}

$$v_{End,i} = \frac{(\frac{v_{End,i-1}}{a_{max}} - \Delta t_{Pred}) + \sqrt{(\Delta t_{Pred} - \frac{v_{End,i-1}}{a_{max}})^2 - \frac{4}{a_{max}} \cdot (\frac{v_{End,i-1}^2}{2 \cdot a_{max}} - \Delta s_{Move,i})}}{\frac{2}{a_{max}}} \quad (10)$$

- If $\Delta s_{Move,i} = 0$ the NC-axis shall be decelerated during the whole SoM i with a_{max}.

In Fig. 11 and Fig 12 the results of this control method are shown for the scenarios 1A and 2B. As it can be seen especially at the profile of the speeds v, this method causes much smoother movements than the method without specific limitations of the speeds.

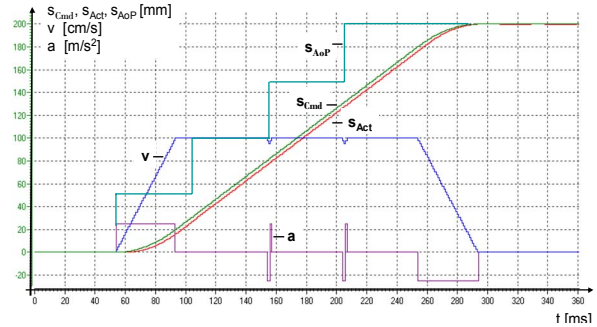

Fig. 11. Movement according scenario 1A with specific limitations of the speeds in each SoM

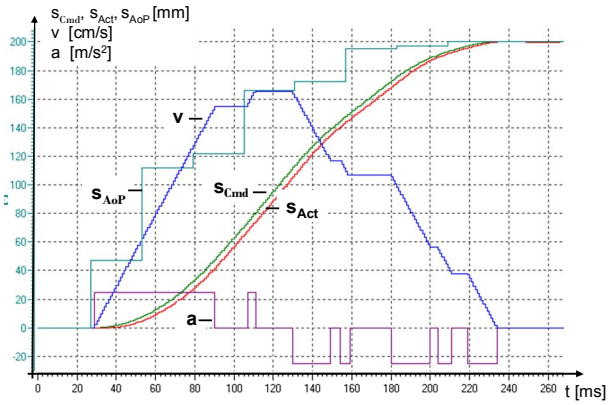

Fig. 12. Movement according scenario 2B with specific limitations of the speeds in each SoM

D. Limitation of the Speeds and the Accelerations

A further improvement can be achieved, if not only the speeds are limited for each SoM but also the accelerations. If a SoM consists either of an acceleration-step and a constant speed or a deceleration-step and a constant speed, the acceleration respectively the deceleration can be distributed to the whole SoM. Therewith the absolute value of the acceleration can be defined smaller than a_{max} and so the mechanical stress to the robot can be reduced further.

In both cases the reduced speed $v_{End,i}$ and the reduced acceleration a_i can be calculated with the following equations:

$$v_{End,i} = (-0{,}5 \cdot \Delta t_{Pred} + \sqrt{(0{,}5 \cdot \Delta t_{Pred})^2 - \frac{2}{a_{max}} \cdot (0{,}5 \cdot v_{End,i-1} - \Delta s_{Move,i})}) \cdot a_{max} \quad (11)$$

$$a_i = \frac{v_{End,i} - v_{End,i-1}}{\Delta t_{Pred}} \quad (12)$$

So all SoM have either a constant speed or only an acceleration-step or only a deceleration-step. The result of this method is shown for scenario 2B in Fig. 13.

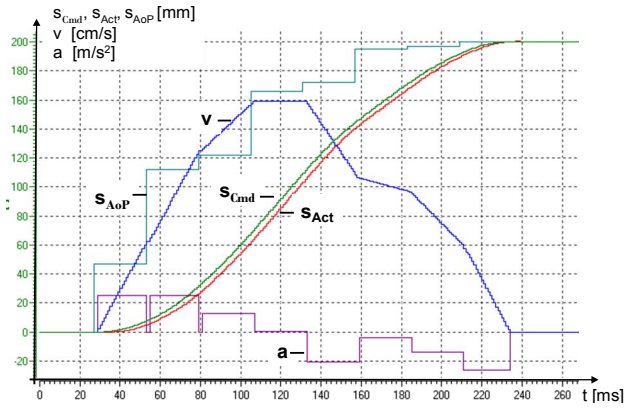

Fig. 13. Movement according scenario 2B with specific limitations of the speeds and the accelerations in each SoM

VII. CONCLUSION

The proposed control methods were tested with a Cartesian Robot. For a movement according to scenario 2B the robot needs with no specific limitations of the speeds and the accelerations about 230 ms (Fig. 10) and with a reduction of the speeds and the accelerations about 235 ms (Fig. 12, Fig. 13). So with the proposed algorithms at a very small loss of time a significant improvement for smooth movements of the robot axis can be achieved.

REFERENCES

[1] H. Frank, N. Wellerdick-Wojtasik, B. Hagebeuker, G. Novak, S. Mahlknecht, "Throwing Objects – A bio-inspired Approach for the Transportation of Parts," in Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, Dec. 17 - 20, 2006, Kunming, China, pp. 91 – 96.

[2] H. Frank, D. Barteit, N. Wellerdick-Wojtasik, T. Frank, G. Novak, S. Mahlknecht, "Autonomus mechanical Controlled Grippers for Capturing Flying Objects," in Proceedings of the 2006 IEEE International Conference on Industrial Informatics, July 3-26, Vienna, Austria, pp. 431 – 436.

[3] N. Furukawa, A. Namiki, S. Taku, M. Ishikawa, "Dynamic Regrasping Using a High-speed Multifingered Hand and a High-speed Vision System," in Proceedings of the 2006 IEEE International Conference of Robotics and Automation, pp.181 – 186, 2006.

[4] U. Frese, B. Bäuml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hähnle, G. Hirzinger, "Off-the-Shelf Vision for a Robotic Ball Catcher," Institute of Robotics and Mechatronics. German Aerospace Center (DLR Oberpfaffenhofen), Available: http://www.robotic.dlr.de

[5] L. Acosta, J.J. Rodrigo, J.A. Mendez, G.N. Marichal, M. Sigut: "Ping-Pong-Player Prototype – A PC-Based, Low-Cost Ping-Pong Robot," in IEEE Robotics and Automation Magazine December, pp. 44 – 52, 2003.

[6] S. Taku, A. Namiki, M. Ishikawa, "Ball Control in High speed Batting Motion using Hybrid Trajectory Generator," in Proceedings of the 2006 IEEE International Conference of Robotics and Automation, pp.181 – 186, 2006.

[7] M. Riley, C. Atkeson, "Robot Catching: Towards Engaging Human-Humanoid Interaction," in Autonom. Robots 12 (2002), pp. 119-128

[8] P. Stiefenhöfer, „Treffsichere Dart-Scheibe: Die Bildverarbeitung im Fokus," in etz Elektrotechn. und Automation 126 (2005) 11, pp 14 – 17.

[9] M. Callegari, F. Cannella, S. Monti, C. Santolini, P. Pagnanelli, "Dynamic Models for the Re-Enegineering of a High-Speed Cartesian Robot," 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings 8-12 July 2001, Como, Italy pp 560-565.