

Trajectory Planning and Feedforward Design for High Performance Motion Systems

Paul Lambrechts, Matthijs Boerlage, Maarten Steinbuch
Faculty of Mechanical Engineering, Control Systems Technology Group
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
P.F.Lambrechts@tue.nl

Abstract—This paper gives an algorithm for fourth order trajectory planning with constrained dynamics for single axis motion control. A model-based feedforward controller is derived that makes full use of these trajectories. Application to industrial high-precision electromechanical motion systems is motivated. Issues like time-optimality, implementation and digitization are considered. Simulation results show superior effectiveness in comparison with rigid-body feedforward.

I. INTRODUCTION

Feedforward control is commonly applied to high performance industrial motion control systems like robots and pick-and-place units. These systems are often embedded in a factory automation scheme, which provides desired motion tasks. Such motion tasks are then transferred to computer hardware dedicated to the control of the system, leaving the details of planning and execution of the motion to this dedicated motion controller.

For simplicity, the trajectory planning and feedforward control are usually done for each actuating device separately, relying on system compensation and feedback control to deal with interactions and non-linearities. Each actuating device is then considered to be acting on a single mass moving along a single degree of freedom. The feedforward control problem is then to generate the force required for acceleration of the mass over the desired trajectory. Conversely, the trajectory should be such that the force is allowable and can be generated by the actuating device. This approach is referred to as 'mass feedforward' or 'rigid-body feedforward'.

The disadvantage of this approach is its dependence on system compensation and feedback control to deal with unmodelled behavior. The resulting problem formulation can be split in two.

- 1) During the trajectory, position errors and feedback control actions can be large, resulting in unallowable velocities and/or accelerations (hence: actuator forces).
- 2) When arriving at the endpoint, the position error is large and it is necessary to introduce a settling time before subsequent actions or motions are allowed.

To improve on this, many academic and practical approaches are possible. These can be categorized in three.

- 1) Smoothing or shaping the trajectory and/or application of force. The result of this can be good, but it

may also lead to a considerable increase in execution time of the trajectory. Various examples can be found in [2], [5], [6], [7], [11], [4].

- 2) Feedforward control based on plant inversion, either by using a more detailed model or by learning its behavior based on measurements. This does not provide an approach for designing a trajectory. Various examples can be found in [1], [3], [8], [9], [10], [13], [14], [15].
- 3) Feedback control and/or system compensation improvement. Obviously, any feedback control design method can be used for this, but trajectory design is again not considered. Some references given above also discuss the effect of feedback control on trajectory following; e.g. see [9], [10], [15].

This paper will provide a method for fourth order trajectory planning and feedforward control that can be used in addition to all of these approaches. After a review of rigid-body feedforward in section II, 'Fourth order feedforward' will be presented in section III. An accurate planning algorithm is given in section IV. The effect of discrete time implementation will be considered in section V. Finally, some simulation results are given in section VI, followed by conclusions in section VII.

II. RIGID-BODY FEEDFORWARD

The specifics of planning a trajectory and calculating a feedforward signal based on rigid-body feedforward are fairly simple and can be found in many commercially available motion control systems. In this section a short review is given as an introduction to a standardized approach to higher order feedforward calculations.

Consider the configuration of figure 1 with m denoting the mass of the motion system, F the force supplied by the actuating device, x the position and k a viscous damping term. Now suppose we have a given bound on acceleration \bar{a} (i.e. a bound on F), and we want to perform a motion over a distance denoted as \bar{x} . Then the shortest time within which the motion can be performed is calculated as:

$$\bar{x} = 2 \times \frac{1}{2} \bar{a} t_{\bar{a}}^2 \Rightarrow t_{\bar{a}} = \sqrt{\frac{\bar{x}}{\bar{a}}} \Rightarrow t_{\bar{x}} = 2t_{\bar{a}} \quad (1)$$

with $t_{\bar{a}}$ denoting the constant acceleration phase duration and $t_{\bar{x}}$ denoting the total trajectory execution time. Hence,

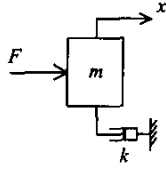


Fig. 1. Simple motion system: a single mass.

the trajectory consists of a constant maximal acceleration phase followed directly by a constant maximal deceleration phase. Clearly if a bound on velocity, denoted as \bar{v} , is taken into account, $t_{\bar{x}}$ can only become larger. We can test whether the velocity bound \bar{v} is violated by calculating the maximal velocity obtained using the minimal time trajectory:

$$\hat{v} := \bar{a} \cdot t_{\bar{a}} \quad (2)$$

Now if $\hat{v} \leq \bar{v}$ we are finished: $t_{\bar{x}} = 2t_{\bar{a}}$ and no constant velocity phase is required. If $\hat{v} > \bar{v}$ we calculate:

$$t_{\bar{a}} = \frac{\bar{v}}{\bar{a}} \Rightarrow x_{\bar{a}} := 2 \times \frac{1}{2} \bar{a} t_{\bar{a}}^2 < \bar{x} \quad (3)$$

and the constant velocity phase duration $t_{\bar{v}}$ is calculated as:

$$t_{\bar{v}} = \frac{(\bar{x} - x_{\bar{a}})}{\bar{v}} \quad (4)$$

resulting in: $t_{\bar{x}} = 2t_{\bar{a}} + t_{\bar{v}}$.

This procedure can now be given as a simple trajectory planning algorithm:

- 1) calculate $t_{\bar{a}}$ from equation 1,
- 2) calculate maximal velocity \hat{v} from equation 2, if $\hat{v} > \bar{v}$: recalculate $t_{\bar{a}}$ from equation 3,
- 3) calculate $x_{\bar{a}}$ from equation 3,
- 4) calculate $t_{\bar{v}}$ from equation 4, and
- 5) finished: $\bar{x} = \bar{a}t_{\bar{a}}^2 + \bar{v}t_{\bar{v}}$ and $t_{\bar{x}} = 2t_{\bar{a}} + t_{\bar{v}}$.

Note that $t_{\bar{v}}$ automatically reverts to zero if the velocity bound is not obtained.

Construction of the acceleration profile a from $t_{\bar{a}}$ and $t_{\bar{v}}$ is straightforward. From this, the desired trajectory can be determined by integrating it once to obtain the velocity profile v , and integrating it twice to obtain the position profile x ; see figure 2. As the position profile thus establishes the trajectory as a sequence of polynomials in time with a degree of at most two, rigid-body feedforward is also referred to as 'second order feedforward'. Note that the feedforward force F is simply calculated from the profiles in figure 2 as:

$$F = ma + kv \quad (5)$$

III. HIGHER ORDER FEEDFORWARD

Compared with the second order trajectory considered in the previous section, higher order trajectories inherently have the advantage of 'smoothing'. This implies a lower energy content at higher frequencies, which results in a lower high frequency content of the error signal, which in turn enables the feedback controller to be more effective.

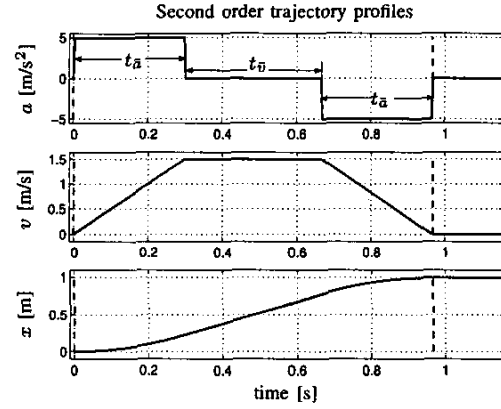


Fig. 2. Second order trajectory determination.

Furthermore this reduces the chance of demanding a motion which is physically impossible to perform by the given motion system: e.g. most power amplifiers exhibit a 'rise time' effect, such that it is impossible to produce a step-like change in force. The result is a decrease of position errors during execution of the trajectory and a reduced settling time. Because of this, many high performance motion systems are already equipped with a third order trajectory planner; in this section it will be determined that a fourth order trajectory planner may give a significant further improvement.

The main argument for this is that an electromechanical motion control system will usually have some compliance between actuator and load, and that both actuator and load will have a relevant mass. For this reason it is natural to extend the single mass model of figure 1 to the double mass model of figure 3. Here m_1 denotes the mass of the actuator, m_2 the mass of the load, F the force supplied by the actuating device, x_1 the actuator position, x_2 the load position, c the stiffness between the two masses, k_{12} the viscous damping between the two masses, k_1 the viscous damping of the actuator towards ground and k_2 the viscous damping of the load towards ground. The equations of

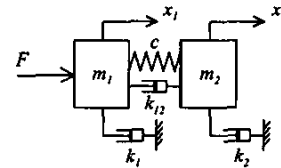


Fig. 3. Extended motion system: double mass.

motion for this configuration are:

$$\begin{cases} m_1 \ddot{x}_1 = -k_1 \dot{x}_1 - c(x_1 - x_2) - k_{12}(\dot{x}_1 - \dot{x}_2) + F \\ m_2 \ddot{x}_2 = -k_2 \dot{x}_2 + c(x_1 - x_2) + k_{12}(\dot{x}_1 - \dot{x}_2) \end{cases} \quad (6)$$

Laplace transformation and substitution then results in:

$$F = \frac{q_1 s^4 + q_2 s^3 + q_3 s^2 + q_4 s}{k_{12} s + c} \cdot x_2 \quad (7)$$

$$\begin{cases} q_1 = m_1 m_2 \\ q_2 = (m_1 + m_2) k_{12} + m_1 k_2 + m_2 k_1 \\ q_3 = (m_1 + m_2) c + k_1 k_2 + (k_1 + k_2) k_{12} \\ q_4 = (k_1 + k_2) c \end{cases}$$

This implies that if we have planned some fourth order trajectory for x_2 , from which we can derive the corresponding profiles for velocity v , acceleration a , jerk j and derivative of jerk d , the feedforward force F can be calculated as:

$$F = \frac{1}{k_{12} s + c} \cdot \{q_1 d + q_2 j + q_3 a + q_4 v\} \quad (8)$$

An implementation of this feedforward scheme is given in figure 4. Analogous to rigid-body feedforward, all required

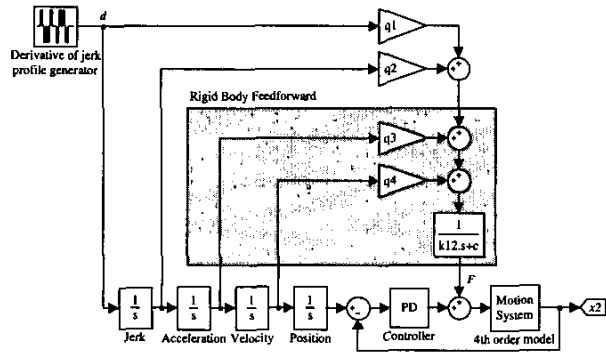


Fig. 4. Fourth order feedforward implementation.

profiles can be obtained by integration of the derivative of jerk profile d . Note that rigid-body feedforward is implicit in this scheme: simply setting $m_2 = k_2 = k_{12} = 0$ makes equation 8 equal to equation 5. The remaining difference is that boundedness of jerk and derivative of jerk will result in a smooth trajectory in comparison with figure 2. This is illustrated in figure 5, which gives an example of a symmetrical fourth order trajectory for a point-to-point move based on the construction of a derivative of jerk profile d . This profile is completely determined by the value of the given bound \bar{d} and the switching time instances $t_0 \dots t_{15}$. An algorithm for obtaining these switching time instances will be the subject of the next section.

IV. FOURTH ORDER TRAJECTORY PLANNING

Planners for second and third order trajectories are fairly well known in industry and academia and there are many approaches for obtaining a valid solution. Extension to fourth order trajectory planning is however not trivial. In this section an approach is given that can be seen as a direct extension of the rigid-body algorithm given in section II.

Assume that the position displacement \bar{x} and bounds on all derivatives of the trajectory up to the derivative of jerk d

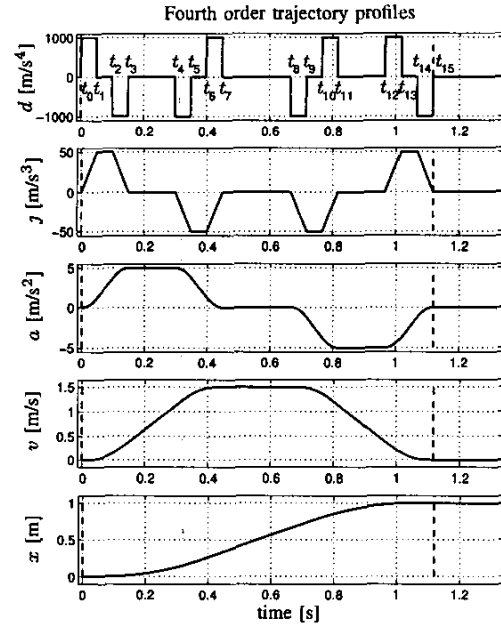


Fig. 5. Fourth order trajectory planning.

are given (indicated as \bar{v} , \bar{a} , \bar{j} and \bar{d}). Furthermore, assume that all derivatives are equal to zero at the start and end positions. The proposed fourth order trajectory planning algorithm is then given in the following steps.

- 1) Temporarily discard \bar{v} , \bar{a} and \bar{j} . From figure 5 follows that d consists of 8 periods with value \bar{d} or $-\bar{d}$.
- 2) Determine $t_{\bar{d}}$: the shortest time of constant d (always first period) such that the total displacement is \bar{x} :

$$t_{\bar{d}} = \sqrt[4]{\frac{\bar{x}}{8\bar{d}}} \quad (9)$$

- 3) Calculate maximal value of velocity \hat{v} :

$$\hat{v} = 2\bar{d}t_{\bar{d}}^3 \quad (10)$$

if $\hat{v} > \bar{v}$: recalculate $t_{\bar{d}}$ based on \bar{v} :

$$t_{\bar{d}} = \sqrt[3]{\frac{\bar{v}}{2\bar{d}}} \quad (11)$$

- 4) Calculate maximal value of acceleration \hat{a} :

$$\hat{a} = \bar{d}t_{\bar{d}}^2 \quad (12)$$

if $\hat{a} > \bar{a}$: recalculate $t_{\bar{d}}$ based on \bar{a} :

$$t_{\bar{d}} = \sqrt{\frac{\bar{a}}{\bar{d}}} \quad (13)$$

- 5) Calculate maximal value of jerk \hat{j} :

$$\hat{j} = \bar{d}t_{\bar{d}} \quad (14)$$

if $\hat{j} > \bar{j}$: recalculate $t_{\bar{d}}$ based on \bar{j} :

$$t_{\bar{d}} = \frac{\bar{j}}{\bar{d}} \quad (15)$$

The resulting $t_{\bar{d}}$ will not be changed anymore.

- 6) Temporarily discard \bar{v} and \bar{a} , but not \bar{j} : extend the trajectory symmetrically with periods of constant j whenever j reaches the value \bar{j} or $-\bar{j}$.
- 7) Determine $t_{\bar{j}}$ such that the total displacement is \bar{x} . This is the positive real solution of the third order polynomial equation:

$$t_{\bar{j}}^3 + (5t_{\bar{d}})t_{\bar{j}}^2 + (8t_{\bar{d}}^2)t_{\bar{j}} + (4t_{\bar{d}}^3 - \frac{\bar{x}}{2\bar{d}t_{\bar{d}}}) = 0 \quad (16)$$

- 8) Calculate maximal value of velocity \hat{v} :

$$\hat{v} = 2\bar{d}t_{\bar{d}}^3 + 3\bar{d}t_{\bar{d}}^2t_{\bar{j}} + \bar{d}t_{\bar{d}}t_{\bar{j}}^2 \quad (17)$$

if $\hat{v} > \bar{v}$: recalculate $t_{\bar{j}}$ based on \bar{v} ; this is the positive real solution of the second order polynomial equation:

$$t_{\bar{j}}^2 + 3t_{\bar{d}}t_{\bar{j}} + 2t_{\bar{d}}^2 - \frac{\bar{v}}{\bar{d}t_{\bar{d}}} = 0 \quad (18)$$

- 9) Calculate maximal value of acceleration \hat{a} :

$$\hat{a} = \bar{d}t_{\bar{d}}^2 + \bar{d}t_{\bar{d}}t_{\bar{j}} \quad (19)$$

if $\hat{a} > \bar{a}$: recalculate $t_{\bar{j}}$ based on \bar{a} :

$$t_{\bar{j}} = \frac{\bar{a}}{\bar{j}} - t_{\bar{d}} \quad (20)$$

The resulting $t_{\bar{j}}$ will not be changed anymore.

- 10) Temporarily discard \bar{v} but not \bar{a} : extend the trajectory symmetrically with periods of constant a whenever a reaches the value \bar{a} or $-\bar{a}$.
- 11) Determine $t_{\bar{a}}$ such that the total displacement is \bar{x} . This is the positive real solution of the second order polynomial equation:

$$\{t_{\bar{d}}^2 + t_{\bar{d}}t_{\bar{j}}\}\bar{d}t_{\bar{a}}^2 + \{6t_{\bar{d}}^3 + 9t_{\bar{d}}^2t_{\bar{j}} + 3t_{\bar{d}}t_{\bar{j}}^2\}\bar{d}t_{\bar{a}} + \{8t_{\bar{d}}^4 + 16t_{\bar{d}}^3t_{\bar{j}} + 10t_{\bar{d}}^2t_{\bar{j}}^2 + 2t_{\bar{d}}t_{\bar{j}}^3\}\bar{d} - \bar{x} = 0 \quad (21)$$

- 12) Calculate maximal value of velocity \hat{v} :

$$\hat{v} = 2\bar{d}t_{\bar{d}}^3 + 3\bar{d}t_{\bar{d}}^2t_{\bar{j}} + \bar{d}t_{\bar{d}}t_{\bar{j}}^2 + \bar{d}t_{\bar{d}}^2t_{\bar{a}} + \bar{d}t_{\bar{d}}t_{\bar{j}}t_{\bar{a}} \quad (22)$$

if $\hat{v} > \bar{v}$: recalculate $t_{\bar{a}}$ based on \bar{v} :

$$t_{\bar{a}} = \frac{\bar{v} - 2\bar{d}t_{\bar{d}}^3 - 3\bar{d}t_{\bar{d}}^2t_{\bar{j}} - \bar{d}t_{\bar{d}}t_{\bar{j}}^2}{\bar{d}t_{\bar{d}}^2 + \bar{d}t_{\bar{d}}t_{\bar{j}}} \quad (23)$$

- 13) Calculate total displacement as if no constant velocity phase is required:

$$x_{\bar{a}} = \{8t_{\bar{d}}^4 + 16t_{\bar{d}}^3t_{\bar{j}} + 10t_{\bar{d}}^2t_{\bar{j}}^2 + 2t_{\bar{d}}t_{\bar{j}}^3 + t_{\bar{d}}^2t_{\bar{a}}^2 + t_{\bar{d}}t_{\bar{j}}t_{\bar{a}}^2 + 6t_{\bar{d}}^3t_{\bar{a}} + 9t_{\bar{d}}^2t_{\bar{j}}t_{\bar{a}} + 3t_{\bar{d}}t_{\bar{j}}^2t_{\bar{a}}\}\bar{d} \quad (24)$$

- 14) Calculate constant speed phase duration $t_{\bar{v}}$ such that the total displacement is \bar{x} :

$$t_{\bar{v}} = \frac{\bar{x} - x_{\bar{a}}}{\bar{v}} \quad (25)$$

- 15) Finished: \bar{d} , $t_{\bar{d}}$, $t_{\bar{j}}$, $t_{\bar{a}}$ and $t_{\bar{v}}$ completely determine the trajectory.

The trajectories resulting from this algorithm have two important inherent properties:

- none of the given bounds is violated,
- in case there is a constant velocity phase ($t_{\bar{v}} > 0$), the trajectory is time-optimal.

Furthermore, although obviously more complex than the rigid-body approach, the algorithm consists of straightforward calculations that can relatively easily be implemented in state-of-the-art motion control hardware.

V. IMPLEMENTATION ASPECTS

This section gives some considerations on the trajectory planning algorithm and the feedforward control scheme of figure 4 for implementation in digital hardware.

A. Switching times

When considering discrete time implementation, the switching time instances of the planned trajectory must be synchronized with the sampling time instances. This implies that the time intervals $t_{\bar{v}}$, $t_{\bar{a}}$, etc. must be rounded-off towards a multiple of the sampling time interval T_s . To remain within the given bounds, but at the same time approximate them as closely as possible, this rounding off must be done to the next higher multiple. The suggested approach is to do this immediately after each calculation of a time interval, after which the maximal value \bar{d} must be recalculated accordingly.

As an example consider the first calculation of $t_{\bar{d}}$ (equation 9). The rounded-off value for $t_{\bar{d}}$ is:

$$t'_{\bar{d}} = \text{ceil}\left(\frac{t_{\bar{d}}}{T_s}\right) \times T_s \quad (26)$$

with $\text{ceil}(\cdot)$ denoting the rounding off towards the next higher integer. From equation 9 we can then calculate a new value for \bar{d} :

$$\bar{d}' = \frac{\bar{x}}{8t'^4_{\bar{d}}} \quad (27)$$

Note that with $t'_{\bar{d}} \geq t_{\bar{d}}$ we must have $\bar{d}' \leq \bar{d}$. It can be verified that this same approach is valid for the calculation or recalculation of all time intervals. Note that with each new calculation of \bar{d}' its value must reduce. This guarantees that none of the bounds that were checked in earlier steps of the algorithm will be violated.

B. Synchronization of profiles

The discrete time implementation of the integrators in figure 4 can be done by replacing the continuous time integrators by forward Euler discrete time integrators as in figure 6. Clearly, all required profiles are now calculated

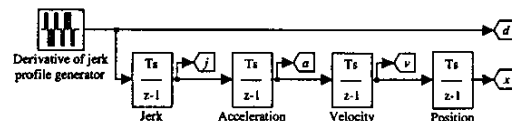


Fig. 6. Discrete time planner using forward Euler integrators.

with sampling time interval T_s . However, due to the zero

order hold effect, each of the four integrators introduces a specific delay time. This can be seen in figure 7a, in which the discrete time profiles are compared with the corresponding continuous time profiles. Note that $T_s = 0.05$ s, which is chosen large in relation to the required trajectory to show the discretization effect more clearly.

To fix this effect, the higher order profiles can be delayed individually such that the symmetry of the complete set of profiles is restored. Figure 7b shows this: the discrete time profiles are now perfectly synchronized with the $2T_s$ delayed continuous time profiles.

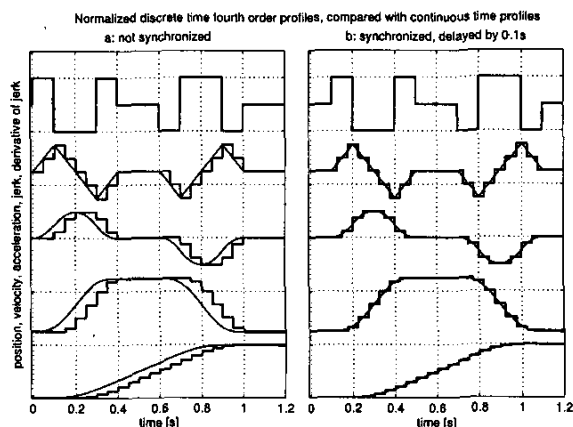


Fig. 7. Discrete time fourth order profiles using forward Euler integrators, compared with equivalent continuous time profiles.

Note that the derivative of jerk profile must be delayed with $2T_s$, the jerk profile with $1\frac{1}{2}T_s$, the acceleration profile with T_s and finally the velocity profile with $\frac{1}{2}T_s$. To obtain a delay of $\frac{1}{2}T_s$ when sampling with T_s the average value is taken from the current and previous amplitude of the considered profile. This operation appears to work very well, although the associated smoothing effect is undesirable.

C. Implementation of first order filter

All required profiles for calculation of the feedforward signal are now available. The multiplication with factors q_1 to q_4 followed by summation as indicated in figure 4 is straightforward. The first order filtering is less trivial, as it must also be transferred to discrete time. A possible implementation that prevents problems with unwanted time delays and gives good results is to make use of the trapezoidal integration method as shown in figure 8.

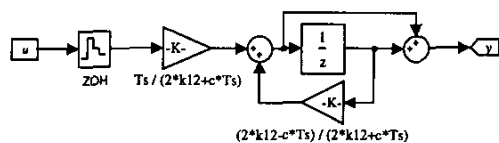


Fig. 8. Discrete implementation of first order filter using the trapezoidal integration method.

D. Calculation of reference trajectory

A final point on synchronization must be made with respect to the calculation of the reference trajectory that is used for feedback control. When applying the synchronized feedforward signal as given above, the actual plant's response will be close to the ideal continuous time response with a delay of $2T_s$. However, in order to compare this response with the reference trajectory it must also be sampled with T_s , leading to an additional delay of $\frac{1}{2}T_s$. Hence, it is necessary to also delay the reference trajectory with this same value.

The result of this is that the control error will not be affected by sampling. The controller will only act on the effects of disturbances and on discrepancies between the actual plant and the modelled fourth order behavior. Obviously, this is only true if the sampling frequency is sufficiently high: otherwise the momentary control error may deviate significantly from the average value. If this is the case, an increase in sampling frequency must be considered. Usually however, the sampling frequency is more significantly determined by the demands on stability and performance of the (digital) feedback controller.

VI. SIMULATION RESULTS

The effects of parameter variations and discretization on the performance of fourth order feedforward control are considered. For this, some simulations are performed using the configuration of figure 4. The motion system parameters and their variations are given in table I. The trajectory

Parameter	Value	Unit	Variation
m_1	20	Kg	$m_1 \in \{15 \dots 25\}$,
m_2	10	Kg	$m_2 = 30 - m_1$
k_1	10	Ns/m	$k_1 \in \{5 \dots 15\}$,
k_2	10	Ns/m	$k_2 = 20 - k_1$
c	$6 \cdot 10^5$	N/m	$\pm 33\%$
k_{12}	500	Ns/m	$\pm 100\%$

TABLE I
SIMULATION PARAMETERS

is calculated for a displacement $\bar{x} = 1$ m, with bounds: $\bar{d} = 1000$ m/s⁴, $\bar{j} = 50$ m/s³, $\bar{a} = 5$ m/s², and $\bar{v} = 1.5$ m/s (see figure 5).

The main concern with model-based feedforward is the discrepancy between the behavior of the actual motion system and the used model. Figure 9 shows the performance of fourth order feedforward for a fourth order motion system model with perturbations according to table I. For comparison, the response of the nominal motion system model with optimal rigid-body feedforward is given (by applying equation 5 to the acceleration and velocity profiles of figure 5). Note that in spite of the significant plant variations, fourth order feedforward performs at least twice as good.

Figure 10 shows that the servo error responses will not significantly deteriorate if fourth order feedforward is implemented in discrete time. As an example, the motion

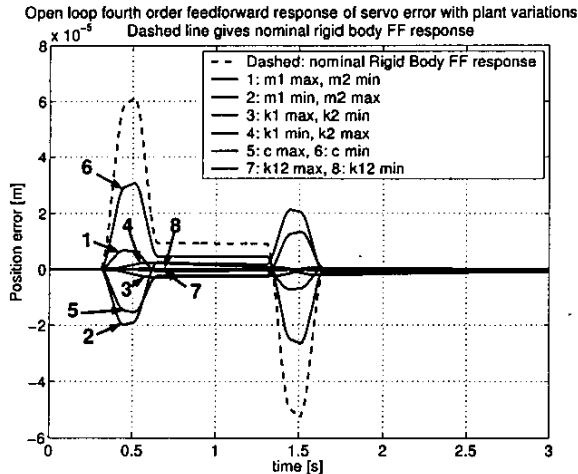


Fig. 9. Open loop simulation results of fourth order feedforward controller with plant variations, in comparison with optimally tuned rigid-body feedforward.

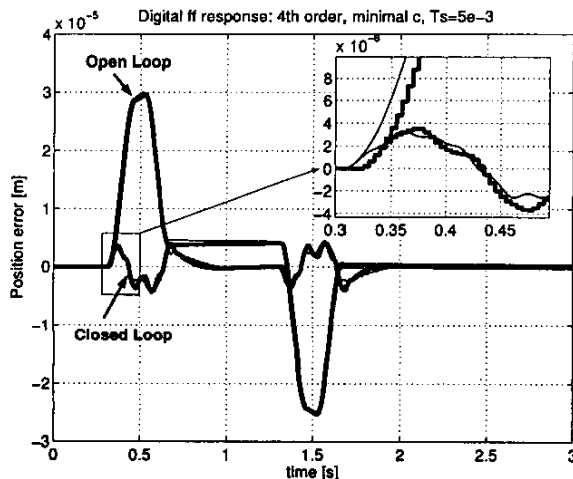


Fig. 10. Simulation results of open loop and closed loop discrete time fourth order feedforward controller with minimal stiffness plant. Thick lines: discrete, thin lines: continuous time.

system with minimal spring-stiffness is considered. Both open loop and closed loop results are given: the feedback controller is tuned for a bandwidth of about 10 Hz, whereas the motion system's first resonance mode is at 50 Hz. The digital feedforward controller is combined with a discrete time feedback controller, both with a sampling rate of 200 Hz. Note that this is a low sampling rate for a high performance servo system: this is chosen to demonstrate the discretization effects more clearly.

VII. CONCLUSIONS

For high performance motion control the usefulness of feedforward is well known. This paper shows that rigid-body feedforward can be extended to fourth order feedforward with superior performance for an important class of motion systems.

An algorithm is given to calculate fourth order trajectories for point-to-point moves with important properties like time-optimality, actuator effort limitation, reliability, implementability and accuracy. Other motion commands, like speed change operations, can be derived from this.

Further implementation issues, like discrete time calculations and synchronization, are addressed. It is shown that deterioration of the continuous time results due to sampling are small when applying a sufficient sampling rate.

REFERENCES

- [1] Devasia S. (2000). Robust inversion-based feedforward controllers for output tracking under plant uncertainty. *Proc. Am. Contr. Conf. 2000*, 497-502.
- [2] Dijkstra B.G., Rambaratsingh N.J., Scherer C., Bosgra O.H., Steinbuch M., & Kersemakers S. (2000). Input design for optimal discrete-time point-to-point motion of an industrial XY positioning table. *Proc. 39th IEEE Conf. on Dec. and Contr.*, 901-906.
- [3] Hunt L., & Meyer G. (1996). Noncausal inverses for linear systems. *IEEE Trans. on Aut. Contr.* 41(4), 608-611.
- [4] Meckl P.H. (1999). Discussion on: comparison of filtering methods for reducing residual vibration. *Eur. J. of Contr.* 5, 219-221.
- [5] Meckl P.H., Arestides P.B., & Woods M.C. (1998). Optimized S-curve motion profiles for minimum residual vibration. *Proc. Am. Contr. Conf. 1998*, 2627-2631.
- [6] Murphy B.R., & Watanaabe I. (1992). Digital shaping filters for reducing machine vibration. *IEEE Trans. on Robotics and Automation* 8(2), 285-289.
- [7] Paganini F., & Giusto A. (1997). Robust synthesis of dynamic prefilters. *Proc. Am. Contr. Conf. 1997*, 1314-1318.
- [8] Park H.S., Chang P.H., & Lee D.Y. (2001). Concurrent design of continuous zero phase error tracking controller and sinusoidal trajectory for improved tracking control. *J. Dyn. Sys., Meas., and Contr.* 5, 3554-3558.
- [9] Roover D. (1997). Motion control for a wafer stage. *Delft University Press*, The Netherlands.
- [10] Roover D., & Sperling F. (1997). Point-to-point control of a high accuracy positioning mechanism. *Proc. Am. Contr. Conf. 1997*, 1350-1354.
- [11] Singer N., Singhose W., & Seering W. (1999). Comparison of filtering methods for reducing residual vibration. *Eur. J. of Contr.* 5, 208-218.
- [12] Steinbuch M., & Norg M.L. (1998). Advanced motion control: an industrial perspective. *Eur. J. of Contr.* 4, 278-293.
- [13] Tomizuka M. (1987). Zero phase error tracking algorithm for digital control. *J. Dyn. Sys., Meas., and Contr.* 109, 65-68.
- [14] Torfs D.E., Swevers J., & De Schutter J. (1991). Quasi-perfect tracking control of non-minimal phase systems. *Proc. 30th Conf. on Dec. and Contr.*, 241-244.
- [15] Torfs D.E., Vuerinckx R., Swevers J., & Schoukens J. (1998). Comparison of two feedforward design methods aiming at accurate trajectory tracking of the end point of a flexible robot arm. *IEEE Trans. on Contr. Sys. Tech.* 6(1), 1-14.