Available online at www.sciencedirect.com

SCIENCE DIRECT®

Control Engineering Practice 13 (2005) 145–157

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Trajectory planning and feedforward design for electromechanical motion systems

Paul Lambrechts*, Matthijs Boerlage, Maarten Steinbuch

Control Systems Technology Group, Faculty of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Received 14 February 2003; accepted 28 February 2004

Abstract

This paper considers trajectory planning with given design constraints and design of a feedforward controller for single-axis motion control. A motivation is given for using fourth-order feedforward with fourth-order trajectories. An algorithm is given for calculating higher-order trajectories with bounds on all considered derivatives for point-to-point moves. It is shown that these trajectories are time-optimal in the most relevant cases. All required equations for fourth-order trajectory planning are derived. Implementation, discretization and quantization effects are considered. Simulations and hardware-in-the-loop experiments show superior effectiveness of fourth-order feedforward in comparison with lower-order feedforward.

Keywords: Trajectory planning; Feedforward compensation; Motion; Point-to-point control; Industrial control; Numerical methods

1. Introduction

Feedforward control is a well-known technique for high-performance motion control problems as found in industry. It is, for instance, widely applied in robots, pick-and-place units and positioning systems. These systems are often embedded in a factory automation scheme, which provides desired motion tasks to the considered system. The current trend is to leave the details of planning and execution of the motion to the computer hardware dedicated to the control of the system: one or more motion controllers. The tasks of such a dedicated motion controller will then consist of:

- *trajectory planning*: the calculation of an allowable trajectory,
- *profile generation*: the representation of the trajectory in appropriate form (e.g. a time sequence with a given sample time),
- *feedforward control*: the calculation of input signals for actuation devices with the intention to obtain the trajectory,
- *system compensation*: to reduce or remove unwanted behaviour like measured disturbances or non-linearities,
- *feedback control*: the processing of available measurements and calculation of input signals for actuation devices to compensate for unknown disturbances and unmodelled behaviour,
- internal checks, diagnostics, safety issues, communication, etc.

This shows that the burden for the motion controller can be quite high, while usually also a high sampling rate is required to achieve the desired performance.

To simplify these tasks, trajectory planning, profile generation 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. In that case, each actuating device is considered to be acting on a simple object, usually a single mass, moving along a single degree of freedom. The feedforward control problem is then to generate the force required to perform the acceleration of the mass in accordance with the desired trajectory. Conversely, the desired trajectory should be such that the required force is allowable (in the sense of mechanical load on the system) and can be generated by

*Corresponding author. Tel.: +31-40-2472839; fax: +31-40-2461418.

E-mail address: p.f.lambrechts@tue.nl (P. Lambrechts).

0967-0661/$ - see front matter
doi:10.1016/j.conengprac.2004.02.010

Nomenclature

$\bar{x}$	bound on $\|x\|$
$\hat{x}$	maximum value obtained by $\|x\|$ if bound is not considered
x_0	initial value
$t_{\bar{x}}$	time interval during which $\|x\|$ obtains its bound
$t_i,\ i \in N$	switching time instances
x	position, displacement (profile) (m or rad)
v	velocity (profile) (m/s or rad/s)
a	acceleration (profile) (m/s^2 or rad/s^2)
j	jerk (profile) (m/s^3 or rad/s^3)
d	derivative of jerk (profile) (m/s^4 or rad/s^4)
F	actuator force, feedforward force (N) or torque (Nm)
m	mass (kg) or inertia (kgm^2)
c	spring stiffness (N/m or Nm/rad)
k	viscous damping coefficient (Ns/m or Ns m/rad)
$q_{1\cdots4}$	feedforward parameters
T_s	sampling time (s)
s	Laplace transform variable
z	shift operator

the actuating device. For obvious reasons, this approach is often referred to as 'mass feedforward' or 'rigid body feedforward'. It allows a simple and practical implementation of trajectory planning and feedforward control.

The disadvantage of this approach is its dependence on feedback control to deal with unmodelled behaviour as mentioned before. The resulting problem formulation can be split into two.

(1) During execution of the trajectory the position errors are large, such that feedback control actions are considerable. Actual velocity and acceleration (hence: actuator force) may therefore be much larger than planned. This may lead to undesired and even dangerous deviations from the planned trajectory and damage to actuator and system.
(2) When arriving at the desired endpoint, the positioning error is large and the dynamical state of the controlled system is not settled. Although the trajectory has finished, it is often necessary to wait for a considerable time before the position error is settled within some given accuracy bounds before subsequent actions or motions are allowed. A practical consequence is the need for a complex test to determine whether settling has sufficiently occurred. Furthermore, it is a source of time uncertainty that may be undesirable on the factory automation level.

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

(1) *Trajectory smoothing or shaping*: This can be done by simply reducing the acceleration and velocity bounds used for trajectory planning, but also by smoothing or shaping the trajectory and/or application of force (higher-order trajectories, S-curves, input shaping, filtering). The result of this can be very good, especially if the dynamical behaviour of the motion system is explicitly taken into account. However, it may also lead to a considerable increase in execution time of the trajectory, often without a clear mechanism for finding a time optimal solution. Various examples of this approach can be found in Dijkstra, Rambaratsingh, Scherer, Bosgra, Steinbuch, and Kerssemakers (2000), Meckl, Arestides, and Woods (1998), Murphy and Watanabe (1992), Paganini and Giusto (1997) and Singer, Singhose, and Seering (1999).
(2) *Feedforward control based on plant inversion*: This attempts to take the effect of unmodelled behaviour into account by either using a more detailed model of the motion system or by learning its behaviour based on measurements. An important practical disadvantage is that they do not provide an approach for designing an appropriate trajectory. Various examples of this can be found in Boerlage, Steinbuch, Lambrechts, and van de Wal (2003), Devasia (2000), Hunt, Meyer, and Su (1996), Park, Chang, and Lee (2001), Roover (1997), Roover and Sperling (1997), Tomizuka (1987), Torfs, Swevers, and De Schutter (1991) and Torfs, Vuerinckx, Swevers, and Schoukens (1998).
(3) *Feedback control optimization* (*possibly aided by system compensation improvement*): By improving the feedback controller, the positioning errors can be kept smaller during and at the end of the trajectory. Furthermore, settling will occur in a shorter time. Also in this case the design of an appropriate trajectory is not considered. Obviously, any feedback control design method can be used for this. Some references given above also include a discussion on the effect of feedback control on trajectory following e.g. see Roover (1997), Roover and Sperling (1997), Torfs et al. (1998).

This paper will provide a method for higher-order trajectory planning that can be used with all of the approaches given above. Furthermore, 'fourth-order feedforward' will be presented as a clear and well implementable extension of 'rigid body feedforward'. It