Chapter 11

Closed-loop, Fieldbus-based Clock Synchronisation for Decentralised Control Systems

Robert Koninckx\(^1\) and Hendrik Van Brussel\(^2\)

\(^1\) Flanders Mechatronics Technology Centre
Celestijnenlaan 300D
B-3001 Leuven (Heverlee), Belgium
bob.koninckx@fmtc.be

\(^2\) Katholieke Universiteit Leuven
Faculty of Engineering, Department of Mechanical Engineering
Division of Production Engineering, Machine Design & Automation
Celestijnenlaan 300B
B-3001 Leuven (Heverlee), Belgium
hendrik.vanbrussel@mech.kuleuven.ac.be

11.1 Introduction

The absolute positioning accuracy of a motion control system is, making abstraction of errors induced by imperfections of the transmission, completely determined by the positioning accuracy of the individual actuators. Accurate contouring additionally requires high degrees of synchronisation between actuators. Indeed, the superposition of two orthogonal and sinusoidal movements only results in a perfect circle if both have exactly the same frequency and if they are out of phase by exactly ninety degrees.

Time synchronisation between actuators is inherently fulfilled as long as all control loops are clocked from the same source. This is evidently the case with the traditional, centralised CNC. However, things completely change for distributed architectures where each individual actuator is equipped with its own processor and oscillator. Even if it were possible to produce quartz crystals with infinite precision, environmental conditions will always lead to small differences between oscillator frequencies and consequently also between the sampling frequencies of the different controllers.

This paper first defines a number of important concepts and terms and summarises the synchronisation requirements of a distributed motion control system. The various approaches to realise time synchronisation of networked computer clocks are subsequently revised. A generally applicable and fully distributed algorithm is finally elaborated. Although this algorithm has only been implemented and tested with the controller area network (CAN), it does
not rely on any specific property of this particular fieldbus. Only guarantees for maximum latencies on message delivery are required, which makes the algorithm usable with any other fieldbus.

11.2 Definition of Concepts

Webster’s Revised Unabridged Dictionary defines the word epoch as “a fixed point of time, established in history by the occurrence of some grand or remarkable event; a point of time marked by an event of great subsequent influence. Epochs mark the beginning of new historical periods, and dates are often numbered from them.” The word is also often used with a similar interpretation in a synchronisation context; the epoch $t$ of events is an abstraction that determines their ordering in a given reference frame or timescale. The epoch thus defines an absolute and universal, but also inaccessible time. A clock is a human artifact to measure the epoch. A “perfect” clock maintains a linear relationship between the epoch $t$ and the measurement $T$ it displays.

$$T(t) = T_0 + R(t - t_0)$$

(1)

$R$ is called the clock’s rate and has the dimensions of a frequency. Equation (1) also shows that the measurement $T(t)$ of the epoch that is displayed by a perfect clock is the integral of a constant frequency $R$, with $T_0$ an integration constant. In other words, the measurement of the epoch that is displayed by a perfect clock represents the phase angle of an oscillator at constant frequency $R$.

$$T(t) = T_0 + \int_{t_0}^{t} R \cdot dt$$

(2)

The term $T_0 + R(t - t_0)$ is for this reason also referred to as the clock’s total phase. Multiplication of this dimensionless number with the inverse of the rate evidently yields an expression in seconds. Since a clock’s total phase equals, up to a scale factor, the time it displays, the remainder of this text does not explicitly differentiate between the two when the context rules out any confusion. Equation (2) also explains why any “real-world” clock consists of an oscillator and a counter. The counter records the number of oscillations since it was initialised with a value $T_0$ at epoch $t_0$, thereby effectively integrating the oscillator’s frequency.

Unfortunately, the ideal behaviour from equation (1) is not realisable in practise and the displayed time will always deviate from the total phase. An extra error term $\epsilon(t)$ must be added to the describing equation to accommodate for inevitable variations on the oscillator’s frequency and the limited accuracy with which the clock can be read [3].

$$T(t) = T_0 + R(t - t_0) + \epsilon(t)$$

(3)