基于 FlexRay 线控转向系统的传输延时算法*

魏志强 张凤登
(上海理工大学 上海 200093)

摘 要：为确保汽车转向系统的实时性，对影响时钟同步的因素——传输延时进行研究，主要针对传输延时的不确定性。通过节点互发同步报文，本地节点动作点获取相应时间戳，并在同步报文发送的负载端里传输时间戳信息，根据时间戳及相应的 FlexRay 节点计算得到传输延时的算法。此算法可以应用于在分布式系统启动时，通过同步报文的传输计算出节点间传播延时和解码消耗后，然后进行配置节点，提高时钟同步精度的同时又不浪费带宽资源。将该算法导入 FlexRay 线控转向系统，并通过 MATLAB 仿真平台对系统进行仿真，得出该算法在对非同步节点进行计算修正之后，使其时钟得到有效调整，提高了系统整体的实时性及可靠性，对汽车技术的发展具有一定意义。

关键词：FlexRay 总线；线控转向；时钟同步；传输延时
中图分类号：TP368.1 **文献标识码**：A **国家标准学科分类代码**：510.0811

Transmission delay algorithm based on FlexRay steering-by-wire system

Wei Zhiqiang Zhang Fengdeng
(University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In order to ensure the real-time performance of vehicle steering system, the transmission delay which is an important factor of clock synchronization is researched in this paper, especially uncertainty of transmission delay. With the nodes send message with each other, action point of local node could get corresponding timestamp message which is sent by the load segment of synchronous message, and the algorithm could be concluded with the timestamp and the corresponding FlexRay nodes. This algorithm could be applied to the startup of distributed system, with transmission of synchronization message, transmission delay and decoding consumption could be calculated, then configuring node, and the clock synchronization accuracy of the system could be improved and the bandwidth resources also couldn't be wasted. According to the simulation of FlexRay steering-by-wire system which the algorithm has been embedded, after correction of transmission delay, the asynchronous nodes could be adjusted effectively. the algorithm improve the overall real-time and reliability, which is of certain significance to the development of automobile technology.

Keywords: FlexRay bus; steering-by-wire; clock synchronization; transmission delay

1 引 言

现如今，随着汽车上的电子技术逐步向自动化和智能化转变，现代汽车电子飞速发展且电子化程度不断深入，其中线控技术(X-by-wire)源于 NASA 在 1972 年最早使用航天科技上的电传操作系统(fly-by-wire)，随后在汽车中得到普遍应用，如：线控转向(steer-by-wire)，线控制动(brake-by-wire)，基于总线设计的车载电动汽车显示装置[1]，基于总线设计的汽车仪表通用终检台[2]等。

FlexRay 总线作为车载总线之一，它的设立是为了“开发面向车内高速控制应用的高级通信技术，提高车辆安全性、可靠性和舒适度”，其基础源于 Daimler Chrysler 的典型应用以及 BMW 公司 Byte flight 通信系统开发的成功经验，能够满足主动安全系统的需要。目前，常用的 CAN 总线虽然在精确时钟同步机制研究方面不断进步[3]，但其在容错功能及带宽方面仍存在不足，而 FlexRay 总线却可以弥补这些不足。因此，汽车领域将 FlexRay 应用于线控系统中[4]。

汽车电子控制系统对实时性要求很高，如：通过速度数据精确测量 GPS 时间延迟[5]，电动汽车 CAN 网络应用层协议的设计[6]，四轮定位仪校准装置关键检测技术[7]以及基于时隙动态分配的 FlexRay 系统通信[8]等。本文的 FlexRay 线控转向系统便属于安全关键性实时系统，用新型的电子控制系统代替传统的机械型和电控液压助力等形

收稿日期：2016-02
* 基金项目：上海市自然科学基金(15ZR1429300)资助项目

式的转向系统。要想保证系统进程之间相互稳定、协调的工作,精确地记录各种事件到达、请求和完成的时间,并且能够保证获得系统精确的全局状态,必须要求有效的时钟同步,而传输延时的改进对确保有效的时钟同步具有重要作用。

2 线控系统架构设计

FlexRay 线控转向系统是分布式实时系统,该系统具有高可靠性,容错性方面则是通过节点冗余实现。系统如图 1 所示。

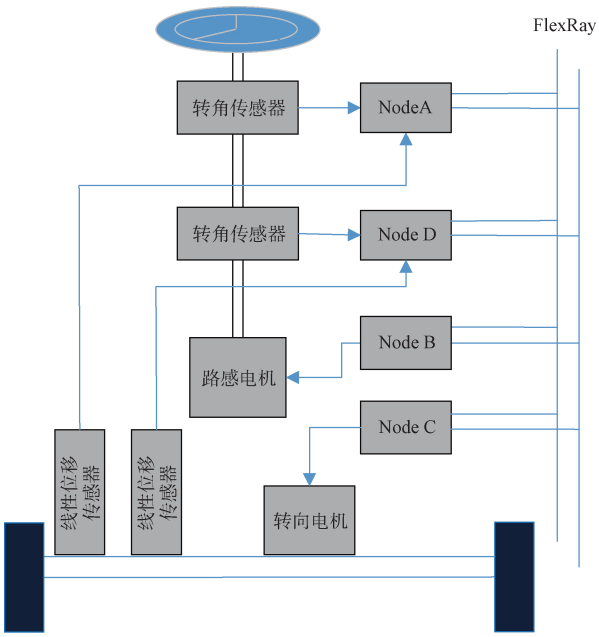

图 1 FlexRay 线控转向系统

系统共设置 A、B、C、D 四个节点。节点 A 主要是通过转角传感器检测方向盘当前位置,并通过线位移传感器获得车轮的当前位置,并通过 FlexRay 总线发送信息给其他节点;节点 B 根据节点 A 的传感器信息通过路感算法进行路感的模拟;节点 C 则控制转向电机实现车轮位置改变;节点 D 为节点 A 的冗余节点,当节点 A 出现故障时,使用 D 节点。该系统需要一个共同的时间基准,才可确保能够完成正确的通信,从而保证事件的一致性。因此,改进传输延迟对提高系统全局时钟同步从而保障该系统的安全性具有重要意义。

3 传输延时

3.1 传输延时来源及影响因素

传输延时是指消息从发送节点发出时刻与到达接收节点时刻之间花费的时间,即消息占据总线完成传输的时间。影响传播时间的因素有很多,例如总线长度、传输速率、消息大小等,并且在传输过程中受外部温度、电压、恶劣通信环境导致的电磁干扰[9]以及实际通信设备物理性能限制等影响,具有不确定性,提高传输延时的计算精度则有助于提高时钟同步的精度[10-11]。

3.2 传输延时的不确定性

在众多影响传播时间的因素当中,传输延时不确定性是影响分布式系统时钟同步中的一个重要因素。全局时钟同步的过程需要系统中节点通过交换带有本地时间戳的同步报文进行时间偏差计算,同步报文在通信网络上的传输需要时间,会有一定的传输延时。并且在实际应用过程中,报文传输有很多不确定性,比如网络堵塞、控制器 CPU 资源分配、调度已经对中断的反应时间都是不固定的,另外报文在发送端发送和接收端接收过程中,协议栈排队、节点间物理媒介长度等因素也有很大的不确定性,因此传输延时存在很大抖动,可以达到几微秒甚至几十微秒[10]。

通常情况下通信延迟时间分为 4 个部分,即生成延迟、队列延迟、传输延迟和接收延迟,如图 2 所示。其中,生成延迟是从应用层组包到传输至 MAC 层的时延,发送节点处理器接收到本节点的请求,到它将准备好的数据写入缓存队列里的时刻,具有高度不确定性;队列延迟是发送端用于争夺总线传输使用权的时间,也即是从消息帧进入发送缓存到消息帧获得总线控制权的时刻,延时可以达到数百毫秒,是报文传输过程中最不确定的因素;传播延迟是从消息帧占据总线到消息帧脱离总线的时刻。接收延迟是从接收端接收的消息从 MAC 层传送到应用程的时间,即消息帧脱离总线到将其中的有效数据提供给接收节点处理器中目标任务的时间,具有高度不确定性,跟协议栈和操作系统有关。

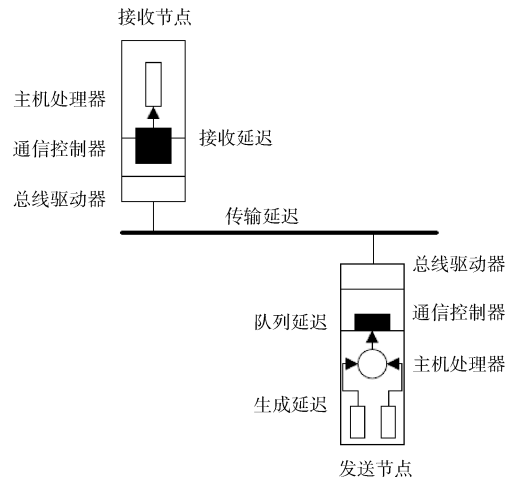

图 2 通信延迟示意图

4 FlexRay 传输延时算法

较常用的时钟同步算法有时钟状态修正的容错中值(fault tolerant midpoint,FTM)算法:FTM 算法不使用节点自身的时间偏差,从偏差序列中去除的最大值和最小值个数 x 是一个系统参数,要根据偏差值的个数来确定,而不

是拜占庭故障数。修正值的计算方法为：首先去除时间偏差序列中的 x 个最大值和 x 个最小值，然后取出剩余序列中的最大值和最小值，将其平均值作为修正值，如表 1 所示。表中 zCorrectValue 为修正值，zlist 为时间偏差由大到小排序后的序列，length 为序列的长度，即偏差值得的个数。FTM 算法表如表 1 所示。

表 1　FTM 算法表

时间偏差值个数	x	修正值计算
1～2	0	zCorrectValue＝(zlist(1)＋zlist(length))/2
3～7	1	zCorrectValue＝(zlist(2)＋zlist(length-1))/2
＞7	2	zCorrectValue＝(zlist(3)＋zlist(length-2))/2

由于这种算法有利于简化硬件设计和克服某些不稳定故障的影响，现已被 FlexRay 总线采用。对 FTM 算法的修正主要是考虑时间偏差测量及对时钟影响因素较大的因素角度方面进行修正。本文通过对传输延时的影响因素(不确定性)研究，在 FlexRay 时钟同步算法的基础上进行改进。

4.1　FlexRay 传输延时理论分析

在 FlexRay 协议中，时钟同步的时钟偏差值计算是基于同步帧的发送触发点时刻在发送节点和接收节点的时间戳，然而接收节点并没有发送节点的触发时间信息，需要通过同步报文到达时间来推测发送端的触发点时刻，如图 3 所示。

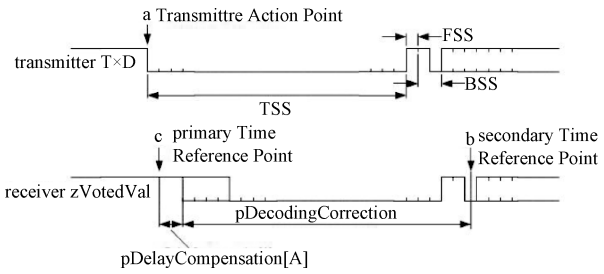

图 3　FlexRay 时钟偏差值计算

FlexRay 接收节点提供了位流定时信息，即发送节点在发送帧数据时，在每字节之前增加了一个由两位组成的字节起始序列(byte start sequence BSS)，接收节点在接收帧期间，将帧的第 1 个 BSS 的第 2 位作为次级时间参考点(secondary time reference point)，当接收节点检测到次级时间参考点时，解码单元采集并储存一个时间戳，记为 zSecondTRP，它与初级时间参考点(primary time reference point)之间的关系为：

$$\text{zPrimaryTRP}=\text{zSecondTRP}-\text{pDecodingCorrection}-\text{pDelayCompensation} \quad (1)$$

式中：pDecodingCorrection 用于修正解码耗时，其取值范围为 14～143 μT，在 FlexRay 协议中 pDecodingCorrection 有相应的计算公式，根据不同的传输速率配置为相应的值，如表 2 所示。

表 2　修正解码耗时计算

BitRate/(Mbit·s^{-1})	2.5	5	10
pDecodingCorrectionMin/μT	24	14	18
pDecodingCorrectionMax/μT	63	87	143

pDelayCompensation 用于修正传播延时，其取值范围为 0～200 μT，根据不同的传输速率配置为相应的值，如表 3 所示。

表 3　修正传播延时计算

BitRate/(Mbit·s^{-1})	2.5	5	10
pDelayCompensationMin/μT	0		
pDelayCompensationMax/μT	50	100	200

FlexRay 时钟同步过程中，发送节点动作点是准备发送同步报文时刻，则不包含生成延时，接收节点的动作点为同步报文的预计到达时间，则不包接收延迟，并且 FlexRay 为确定性通信，静态段采用时间触发通信，节点发送和接收在哪个时隙都是提前配置好的，不会存在网络消息堵塞，即没有队列延迟。在时钟偏差计算中用到的次级时间参考点 zSecondTRP，在推测同步消息的实际到达时间时用到了解码修正量 pDecodingCorrection 和传播延时修正量 pDelayCompensation，在 FlexRay 协议在中对于这两个值的配置如表 2 和 3，这两个值的配置影响时钟同步精度。

4.2　FlexRay 传输延时算法

针对传输延时不确定性，最常用解决办法是往返测量法，通过时间报文在节点间来回发送，节点利用请求-相应处理查询另一个节点的时钟状态。如图 4 所示，节点 A 在时间 T_1 发送一个请求给节点给 B，B 会根据自己的实际记录接收时间 T_2，并在 T_3 返回一个响应报文，T_3 比 T_2 晚。最后，A 记录下响应报文到达时间 T_4。

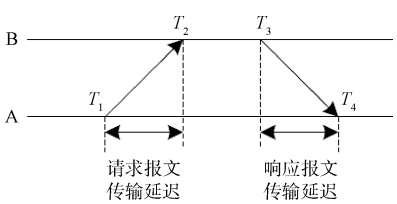

图 4　从另一个节点得到当前时间

则有:

$T_2-T_1=T(A-B)+Toffset$ (2)

$T_4-T_3=T(B-A)-Toffset$ (3)

假设节点 A 与 B 之间的网络传输延时是对称的,则从 A 到 B 的传输延时 $T(A-B)$与从 B 到 A 的 $T(B-A)$大致相同,即:

$Tdelay=T(A-B)=TB-A$ (4)

这样,A 就可以计算出与 B 的时间偏差:

$Toffset=((T_2-T_1)-(T_4-T_3))/2$ (5)

A 到 B 的传输延时为:

$Tdelay=T(A-B)=T(B-A)=((T_2-T_1)+(T_4-T_3))/2$ (6)

这种计算方法可以采用多次重复测量计算出传输延迟,能够提高系统时钟同步精度,但是它是在假设 A 与 B 之间的网络传输延时对称的前提下计算的,对于一些网络对称性能低的系统,该方法提高时钟同步精度的作用将会大大减小。

上节中介绍了 FlexRay 时钟同步过程在计算偏差测量值的时候,对同步报文由发送节点到达接受节点间的传播延时和解码耗时进行了修正,协议中对于这两个值的配置只是根据总线速率定义了最大值最小值。然而影响传播延时大小的因素不仅仅有传输速率,还有消息大小、导线长度、外部环境等,这样会存在一定误差,影响安全关键性系统时钟同步精度。

对于 FlexRay 网络,其消息传输具有确定性,报文长度已经报文传输时隙都是提前配置好的。在线控转向系统中相邻两个节点的节点采用相同性能和参数的控制器,则同步消息在节点间来回传输同步报文时,传播延时和解码耗时具有对称性,则可以使用往返测量法对其进行测量。

对于线控转向系统中 A、B、C、D 四个同步节点,以测量 A、B 间传播延时和解码耗时为例,其他节点间方法类似。

定义节点 A 在时隙 1 发送同步报文,节点 B 接收报文,收到报文后记下。

$ZSecondTRP_{A-B}$和 $zActionPoint_{A-B}$时间戳;节点 B 在时隙 2 发送同步报文,并把节点 B 在时隙 1 接收同步报文产生的 $ZSecondTRP_{A-B}$和 $zActionPoint_{A-B}$时间戳放在负载端,一起发送给节点 A,节点 A 接收同步报文,记下 $ZSecondTRP_{B-A}$和 $zActionPoint_{B-A}$时间戳,并读取负载端的内容;则会有相应的时间戳及相位偏差关系:

$DevValue_{AB}=zPrimaryTRP_{A-B}-zActionPoint_{A-B}=zSecondTRP_{A-B}-pDecodingCorrection_B-pDelayCompensation_{A-B}-zActionPoint_{A-B}$ (7)

$DevValue_{BA}=zPrimaryTRP_{B-A}-zActionPoint_{B-A}=zSecondTRP_{B-A}-pDecodingCorrection_A-pDelayCompensation_{B-A}-zActionPoint_{B-A}$ (8)

式中:

$DevValue_{AB}=-DevValue_{BA}$

$pDelayCompensation_{A-B}=pDelayCompensation_{B-A}=pDelayCompensation$

$pDecodingCorrection_A=pDecodingCorrection_B=pDecodingCorrection$

则可得到:

$pDelayCompensation+pDecodingCorrection=((zSecondTRP_{A-B}-zActionPoint_{A-B})+(zSecondTRP_{B-A}-zActionPoint_{B-A}))/2$ (9)

节点 A 通过时隙 1 和 2 的报文,就能计算出节点 A 与节点 B 之间的传播延时和解码耗时之和。并且可以采取多次测量,求平均值,使用概率技术减少测量过程中的误差,提高测量精度。

这种算法虽然可以提高测量精度,但是节点通信使用了大量的负载段信息,并且一旦决定了传播延时和解码耗时配置值就无需再进行计算,所以此算法可以应用于在分布式系统启动时,通过同步报文的传输计算出节点间传播延时和解码耗后,然后进行配置节点,提高时钟同步精度的同时又不浪费带宽资源。

5 实验结果

在完成算法的分析研究之后,通过 MATLAB 仿真平台对系统进行仿真。首先,对于未进行时钟修正时,各节点的时钟运行情况和同步发送情况如图 5 所示,记值为测量段的结束时刻。

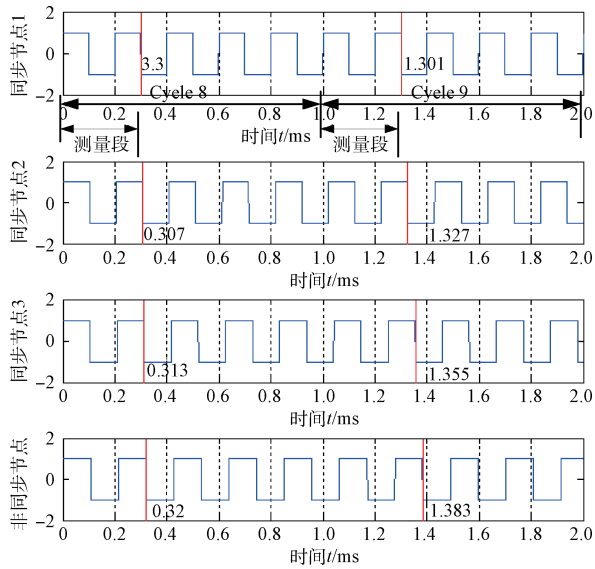

图 5 时钟未修正图

将传输延时算法导入系统后(并开始仿真),可以得到如图 6 所示的修正结果,从图中可以看出,经过一段时间的运行,系统中的非同步节点在计算修正值后进行了时钟调整。说明改进后的传输延时算法可以将不同步的时钟调整

为与同步时钟差距不大的同步时钟。

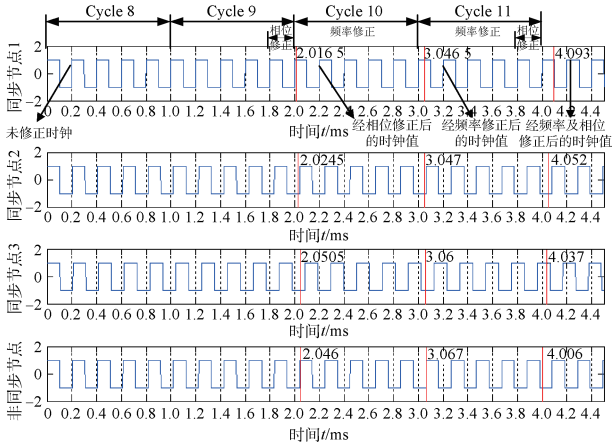

图 6　时钟修正图

由于 FlexRay 采用了分布式的时钟机制，即每个节点通过监视其他节点发送同步帧的时间使自己与簇同步。因此，保证网络中同步节点的性能可以提高整个网络的时钟同步精度，使各节点的时钟修正值更加准确。

6　总　　结

FlexRay 时钟同步是一个非常重要的过程，它的精度影响系统的实时性。本章通过研究影响分布式时钟同步精度的重要因素-传输延时，分析了传输延时的不确定，然后结合 FlexRay 时钟同步机制提出了传输延时算法。传输延时算法通过节点互发同步报文，本地节点动作点获取相应时间戳，并在同步报文发送的负载端里传输时间戳信息，根据时间戳及相应的 FlexRay 节点计算传输延时的公式，从而得到 FlexRay 传输延时算法。最后利用 MATLAB 仿真工具对系统进行仿真，由仿真结果分析可知，将算法应用于系统中可有效提高系统时钟同步的精度，进而提高了系统的实时性及可靠性，对汽车技术的发展有一定意义。

参考文献

[1]　黄得铭，武鹏. 基于柔性测试技术的车载电动机车数据显示装置设计[J]. 电子测量技术，2012，35(2)：5-8.

[2]　邢益临；陈洪雨；陈波等. 汽车仪表通用终检台的设计与实现[J]. 电子测量技术，2012，35(7)：84-87.

[3]　王跃飞，杨锦，张利，等. 汽车 CAN 系统精确时钟同步机制研究[J]. 电子测量与仪器学报，2014，28(1)：22-28.

[4]　顾嫣. FlexRay 线控转向系统的实时性与容错性研究[D]. 上海：上海理工大学，2010.

[5]　伍洪俊，张辉，庄儒耀，等. 基于速度数据的 GPS 时间延迟精确测量方法[J]. 电子测量技术，2014，37(11)：106-108.

[6]　卫星，张建军，张利，等. 电动汽车 CAN 网络应用层协议研究[J]. 电子测量与仪器学报，2011，25(9)：799-804.

[7]　张起勋，邵承会，张忠元，等. 3D 四轮定位仪校准装置关键检测技术研究[J]. 仪器仪表学报，2014，35(9)：1979-1989.

[8]　王跃飞，曹三峰，毕翔，等. 一种基于时隙动态分配的 FlexRay 系统通信机制[J]. 电子测量与仪器学报，2015，29(2)：179-186.

[9]　孙文超，毛征，张辉，等. 基于 2B+D 通信系统时延测算的实现[J]. 国外电子测量技术，2015，34(2)：39-42.

[10]　何建平. 基于一致性的无线传感器网络时钟同步算法研究[D]. 杭州：浙江大学，2013.

[11]　陈珍萍，李德权，黄友锐，等. 无线传感器网络混合触发一致性时间同步[J]. 仪器仪表学报，2015，36(10)：2193-2199.

作者简介

魏志强，1993 年生出，在读硕士研究生，主要从事现场总线、汽车电子方向的研究。

E-mail：zqweifly@163.com

张凤登，1963 年出生，工学博士，教授，硕士生导师，主要研究方向为现场总线、汽车电子。

E-mail：FDZhang@usst.edu.cn