

Development and Implementation of CANopen based Controller for Skid Steer In-Wheel Drive Vehicle

Priya R
Junior Research Fellow,
R & DE Establishment, Pune.

Chandrajit Ganguly
Scientist D,
R & DE Establishment, Pune.

Abstract:- The Controller Area Network (CAN) is one of the widely used protocol for intra-vehicle communication between Electronic Control Units (ECUs) which broadens its range of applications for industrial automation field. This paper demonstrates a software framework for motion control performance based on CANopen network emphasizing on testing and verification of systems with a table top setup which can be implemented in an early development phase before the integration. The CAN module delivers modest hardware interface in addition to highly reactive software communication protocol, and therefore can be effortlessly embedded into any systems. Service Data Object (SDO) is chosen as the communication object in CANopen protocol between NUVO-VTC 5104 along with multiple motor drives. Simulation is performed with Visual Studio in windows as well as Linux platform whereas CANopen communication with physical components is realized via stubs written in C++. The aids of multi-axis control with the CANopen protocol for multiple motor drives are higher data rate, reliable transmission of data, stabilization as well as effortless communication topology.

Keywords — CANopen, Lightweight Communications and Marshalling (LCM), Motion Controller, Hardware-in-loop simulation (HILS) NUVO-VTC 5104, Logitech F710 Controller.

I. INTRODUCTION

Controller Area Network (CAN) is a serial communication network which efficiently supports distributed as well as real-time control, broadening the applications in the field of automation with high performance and high reliability. Vehicle control systems widely use embedded systems providing intelligent control systems for upscale cars. In order to make sure that the CAN bus is reliable, rigorous application layer protocol must be framed. Arbitration mechanism is used for media access control which confirms that transformation of messages are carried out based on the priority allocated.

CANopen protocol is one of the standards for CAN-in-Automation (CiA) definition, a high-level agreement which bases on CAN serial bus system and the application layer. CANopen protocol delivers the required procedure for distributed control as well as embedded system such as interoperability and interchangeability between different CAN equipment, standardized and unified system communication mode, device description way and network function, and network node functions of any expansion.

CANopen is an open standard with highly flexible

configuration capabilities developed for motion-oriented applications. Currently it is used in several domains, such as medical equipment, off-road vehicles, maritime or railway applications. CANopen protocol consists of six core concepts which is discussed in the section II, one of them is "Object Dictionary". Object Dictionary plays a vital role in this protocol which acts as a database for the features applicable in communicating with the devices or the application along with the concerned data in the communication.

CANopen components are mainly used for machinery and ships, and therefore they are part of highly individually planned and implemented systems. Thus, simulation plays a key role in system design, including the modeling of the user's behavior and the environment. CAN communication has a wider set of applications due to its features such as highly reliable, robustness, modest growth, cost effective, short frame transmission and non-destructive arbitration technology, etc. Nevertheless, CAN defines the physical and the data link layer, with no traces of application layer. Therefore, CANopen protocol approaches in order to eliminate this gap between the application layer and the user.

CANopen in motion control for communication can not only resolve difficulties with the connection, but also increase the reliability of the information transmission, instant response, tuning etc. A collective series of recent engineering usages requires great communication interface strength as well as the time consumed within the nodes involved in a technical procedure. The same is applicable with multi-axis motion control systems, where the accuracy of the axis as well as the synchronization within them is to be maintained precise.

The need of work synchronization amongst several servo drives, sensors along with several connected devices requires a good communication interface and compatibility along with data throughput which can be achieved through the designed protocol.

The paper is organized as follows. The next section presents the implementation of the CANopen protocol along with the system overview and its functionality. The subsequent section details with the table top set up, hardware-in-loop simulation, and control algorithms for system evaluation. Section V details the results of the simulation analysis and concludes the paper, making recommendations and suggestions for enhancement.

II. IMPLEMENTATION OF CANOPEN

A. Structure of CANopen

CAN is used to communicate between electronic controls unit on automobiles. Data is transferred through different nodes and control units reducing the use of sensor integration and cabling.

Standard Data Frame

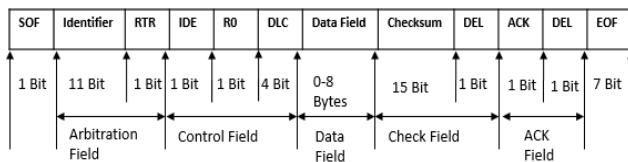


Figure 1 Standard CAN Data Frame

CAN uses arbitration field, control field and data field to interact with the user. Message priority is based on the content of the arbitration field where the messages are prioritized as high when the field content is zero making sure that higher priority messages are transmitted first. This can also be used to determine the time taken to transmit a CAN message. Based on the type of application, content in the data field can be manipulated.

CANopen is a higher layer protocol based on CiA providing flexibility to configure wide range of applications.

Table 1 CANopen Frame

FUNCTION CODE	NODE ID	RTR	DATA LENGTH	DATA
4 BIT	7 BIT	1 BIT	4 BIT	64 BIT

Majority of the CANopen data will be SDOs (Service Data Objects) and PDOs (Process Data Objects) i.e. information such as torque, position, velocity etc.

B. Six Core Concepts of CANopen

Communication Models - Master/slave, client/server and producer/consumer.

Communication Protocols - Related to the models are various protocols used for different communication purposes - e.g. transmitting real-time data.

States - Each device supports different states. Using particular messages, a "master" node will be responsible for the state of a "slave" node - e.g. resetting it.

Object Dictionary (OD) - Each and every device has its object dictionary, which includes records that are useful for the device configuration and many such initial operations which is accessible through the SDO service.

Electronic Data Sheet (EDS) - It is a standard file format used for the object dictionary which permits the services to update devices without difficulty.

Standardized Device Profiles - CANopen delivers multiple profiles for e.g. input/output (I/O) modules in CiA 401 and motion-control in CiA 402 - to facilitate the ease for the vendor's individuality.

C. System Architecture

This section presents the design of the motion controller system. Figure 2 demonstrates the overview of the system

for motion controller. In this scenario multiple motor drives are connected via the CAN to ensure the communication. These drives are equipped with a built in CAN controller which helps in communicating with the embedded system for the user application.

In this application single embedded controller is used as master CAN controller to interface with the several motor drives that is NUVO- VTC 5104 with built in CAN interface. Motor drives are controlled by interfacing the Bluetooth device with the controller for communicating based on the system requirements with the help LCM (Lightweight Communications and Marshalling) platform. LOGITECH F710 is used as the Bluetooth device which is interfaced with the CAN master for motion controller application. Based on the input provided with the Bluetooth device motor drives responds accordingly via the CAN master.

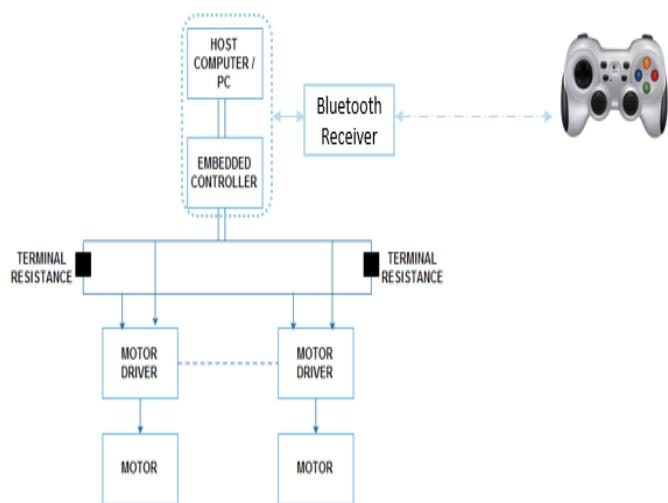


Figure 2 System Overview

D. System Functionality

The system schematic includes a control unit, and electric motors. Software for controlling motors includes multiple tasks. Where specifying the Baud Rate and Node ID is a onetime process which is set using the respective Motor Drive software. The motor drives inherently assign higher priority levels to the CAN tasks. One of the important task is to make sure that the devices has not entered into any of the error states. The factors for controlling the motors depends on the input received from the Bluetooth device which can also be altered based on the requirements.

The key notion in controlling the multiple motor drives using a single embedded controller is based on the different profiles supported by the CANopen protocol that is position, velocity and torque profile. All the tasks are performed once the initial settings are carried out for establishing the network interface between the controller and the motor drives. The software flow for network interface is as shown in figure 3.

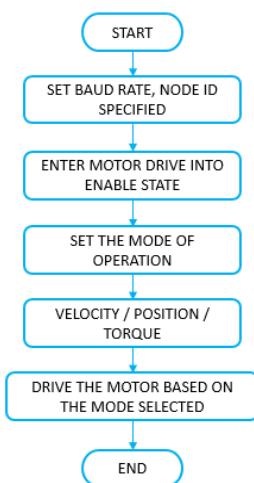


Figure 3 Software Flow Control Flowchart

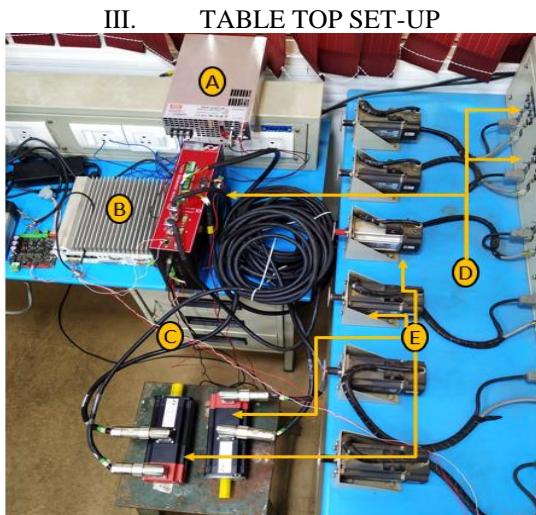


Figure 4 Table top setup of the drives and motors used

A. Power Supply B. NUVO VTC 5104 in vehicle Embedded Controller C. CAN Interface D. Drives used E. Motors used

Figure 4 depicts the table top setup for the experiment conducted where a single embedded controller is interfaced with multiple motor drives.

IV. CONTROL ALGORITHMS FOR SYSTEM EVALUATION

Once the drive parameters are decided and the desired profile is set for the application, the motor drives are controlled and evaluated. Requirements to initiate skid and differential velocity for the drives are controlled based on the Bluetooth leveler position. This mechanism is illustrated in the following flowchart as shown in Figure 5.

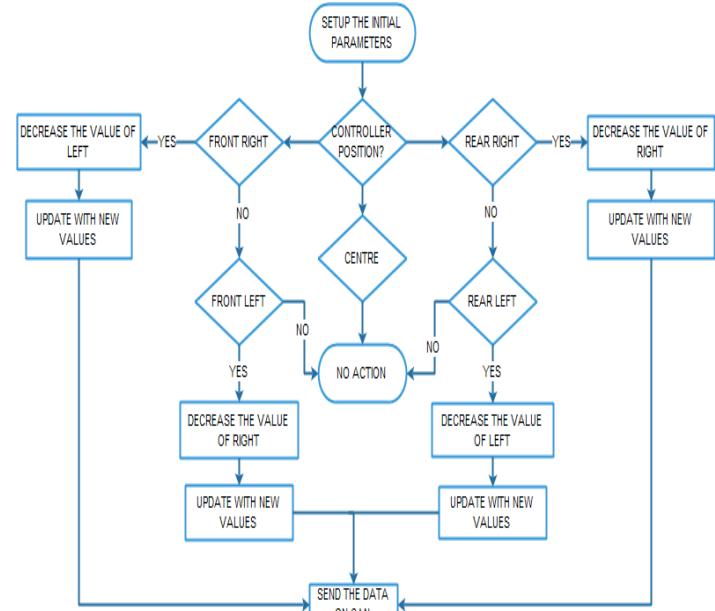


Figure 5 Control Algorithm for Differential Skid Steer

V. RESULTS AND CONCLUSION

The resulting hardware-in-loop simulation model of the system includes three subsystems which are organized thoroughly with regard to the mechanisms of the electric drive. From the results obtained the functionality of the drives involving in the distribution of velocity to each and every motor is made evident from figure 6 and figure 7. The minute the drives observe any skip is indicated in the profile. Motor Drives immediately reacts by the required action that is active distribution in addition the amount of rotation is improved by a part submissive by decreasing the velocity for the required motor drives.

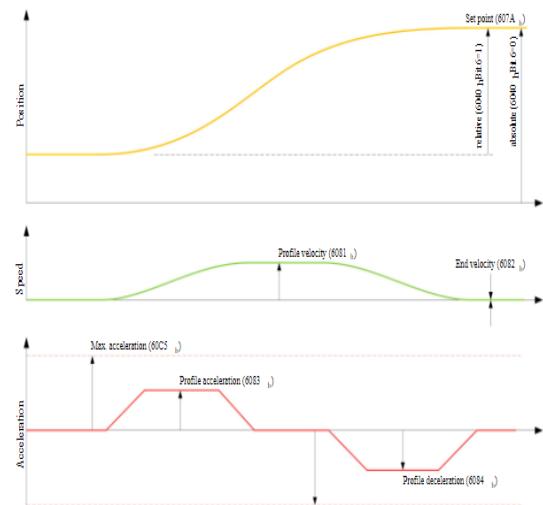


Figure 6 Motion Profile for position and velocity

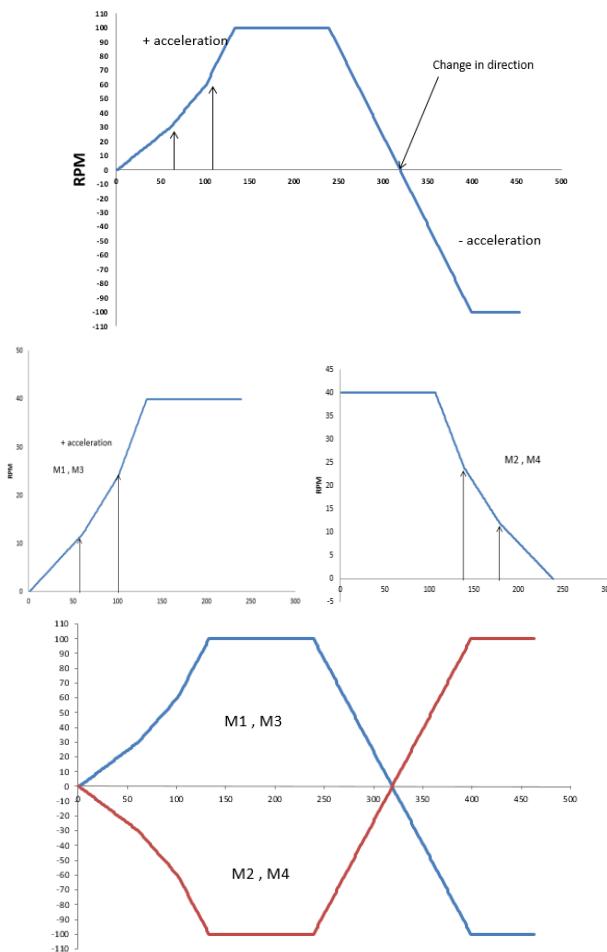


Figure 7 Experimental results of the system

Figure 7 shows the experimental results obtained from the proposed algorithm. Where first figure from top indicates the forward and reverse motion profile of the vehicle, second figure from top indicates the differential mechanism of the drive and the third figure indicates the skid steer mechanism.

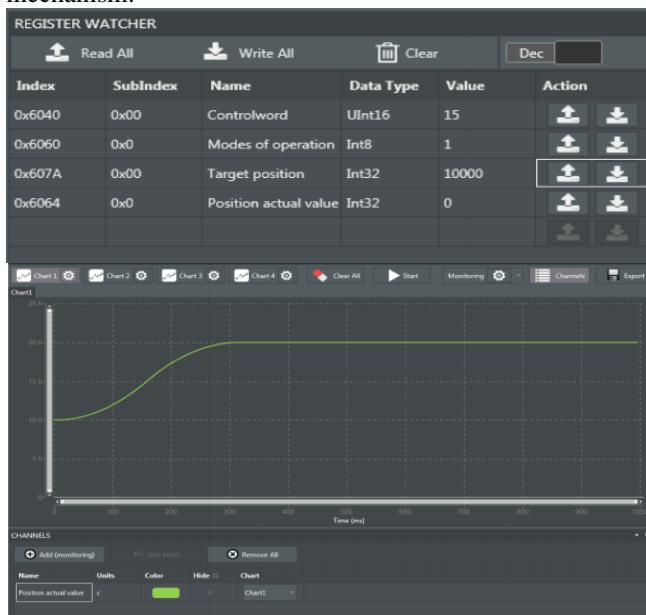


Figure 8 Motion Profile for position mode

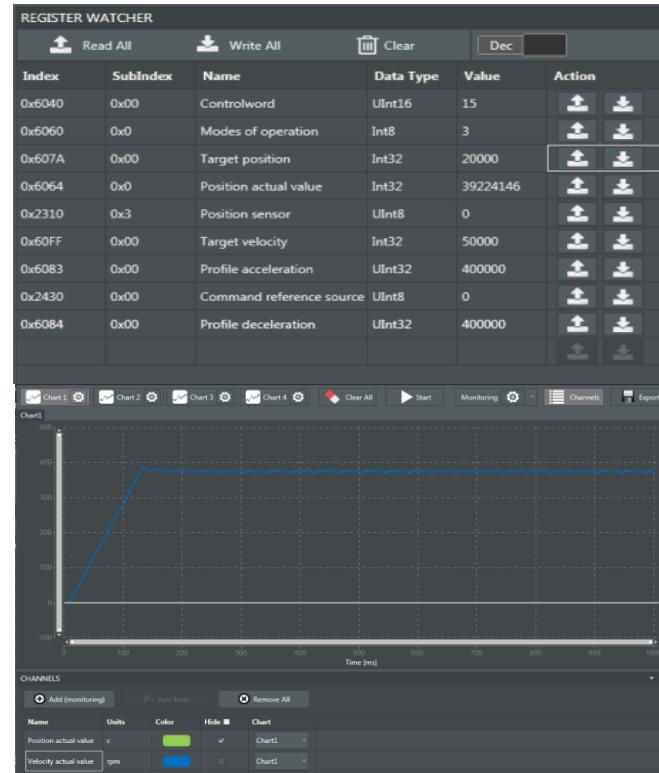


Figure 9 Motion Profile for velocity mode

The proper functionality of the application software for the drives has been verified. The results obtained depicts the accurate functioning of the devices along with its significant uses in operating any of the vehicle providing completely functional control algorithms along with the application software for the motion controller. It also serves as a podium for additional progress in the algorithms for controlling the drives. Altogether the aforementioned application functionalities is tested and proved.

Based on the results of the simulation presented, it is evident that it may serve as a good starting point for implementing this protocol exclusively for communication. As a continuation of this research, we propose to test the performance similarly for more varied types of messages and hardware. Also, the latency of message communication merits further investigation.

REFERENCES

- [1] Yukang Chang, Zhiyong Zhang, and Chunlei Wang "Control System Design Based on CANopen Network for a Transformable Multimodal Robot", Proceedings of 2018 IEEE International Conference on Mechatronics and Automation.
- [2] Nico Submann and Ansgar Meroth, "Model based development and verification of CANopen components", IEEE 2017.
- [3] Chao Zhou, Feng Luo, "Design of the redundant CAN Network based on CANopen", IEEE 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).
- [4] T.Vaishnavi, Dr.Sheeba Joice.C, "Review on Range Of Controller Area Network (Can) Protocol In Different Automobile Appliances", ISSN: 1314-3395, International Journal of Pure and Applied Mathematics.
- [5] A. A. Salunkhe, Pravin P Kamble, Rohit Jadhav, "Design And Implementation of CAN bus Protocol for Monitoring Vehicle Parameters", IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016

- [6] Hamid Rahmani, Karolin Loser, Richard Thum, Ali Hayek, and Josef Borcsok, "CANopen Safety on Safety-Related System-on-Chip", IEEE International Conference on Information, Communication and Automation Technologies (ICAT), 2015
- [7] Aibing Ye, Huayao Zheng, "The Design and Implementation of CANopen Protocol Intelligent Analysis System", IEEE, 2009
- [8] Manuel B. M. Barbosa, Adriano da Silva Carvalho, Mohammed Farsi, "A CANopen 110 Module: Simple and Efficient System Integration", IEEE
- [9] Gianluca Cena, Adriano Valenzano, "Efficient Polling of Devices in CANopen Networks", IEEE 2003
- [10] Luis Vazquez, Leoandro Rojas, Samuel Galceran, and Antoni Sudria, "Simplified CANopen Application Layer Model for Educational Proposals", IEEE, 2001