ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

AGRIBOT - MOBILE ROBOT TO SUPPORT OF AGRICULTURAL PRECISION ACTIVITIES

Jhon F. Archila Diaz
Henry Borrero Guerrero
University of São Paulo, Av. Trabalhador São-carlense, 400, São Carlos, São Paulo, Brazil.
john.faber@usp.br, h_borrelo@ieee.org

Alexander J. Tiberti
Rubens A. Tabile
Giovana Tripoloni Tangerino
Clayton José Torres
Rafael Vieira de Souza
Marcelo Becker
Arthur Jose Vieira Porto
Mario Tronco
University of São Paulo, Av. Trabalhador São-carlense, 400, São Carlos, São Paulo, Brazil.
atiberti@sc.usp.br, rubens.tabile@gmail.com, giovanatt@gmail.com, ctorres@sc.usp.br, rafael.sousa@gmail.com, becker@sc.usp.br, ajvporto@sc.usp.br, mltronco@sc.usp.br

Ricardo Yassushi Inamasu
Embrapa - Brazilian Agricultural Instrumentation Research Corporation, São Carlos, São Paulo, Brazil.
ricardo@cnpdia.embrapa.br.

Abstract. *AgriBOT is a 4WSD (Four Wheel Steering and Driven) vehicle designed and constructed to be robotized and therefore to support precision agriculture activities. AgriBOT allows to research in a diversity of fields from academicals sub-projects with actuation of USP - EESC (Universidade de Sao Paulo Escola de Engenharia de Sao Carlos) graduation students and the participation of important Brazilian entities as FINEP ("Financiadora de Estudos e Projetos"), Embrapa ("Empresa Brasileira de Pesquisa Agropequaria") and Jacto S.A company. A crucial attribute of the robotic platform is the mobility capability which is provided by a four wheel module system, each one of the four wheel modules are identical and provides a system with two-degree-of-freedom system. Each module includes a hydraulic motor for propulsion system and a hydraulic cylinder dedicated to steering. The vehicle has incorporated distributed electronic devices to works as interface between electromechanical actuators and controllers using the Controller Area Network (CAN) protocol. Currently the project has software and hardware applications implemented to control the steering and propulsion subsystems.*

Keywords: *Four Wheel, steering, agricultural vehicle, Controller Area Network, mobile robot.*

1. INTRODUCTION

The mobile robot of AgriBot project, in fact named AgriBot, is a vehicle oriented to support agricultural activities and it is a prototype that allow to research a diversity of fields from academicals sub-projects with the actuation of graduation students of USP - EESC (Universidade de Sao Paulo Escola de Engenharia de Sao Carlos) and the participation of important Brazilian entities as FINEP ("Financiadora de Estudos e Projetos"), Embrapa ("Empresa Brasileira de Pesquisa Agropequaria") and Jacto S.A company. So, AgriBOT is a 4WSD vehicle because the basis for the robotic platform is the mobility capability provided by four wheel module mechanism that are powered by a diesel engine, each one of the four identical wheel modules includes a hydraulic motor actuator for propulsion and other hydraulic actuator for steering, these actuators provides direct drive to create a two-degree-of-freedom system for each module. Communication between electronics devices and electromechanical actuators is made using Controller Area Network (CAN) protocol that is a serial network technology used to communicate Electronic Control Unit (ECU) and was designed for the automotive industry, but many popular in industrial automation as well as other applications. By using 4WSD configuration, it is possible to produce parallel displacement of the vehicle during turns because it decouples adjustments in position from adjustments in orientation. It also allows both the front and rear of the vehicle to follow a specific path precisely and may maintain a fixed orientation relative to the crop rows. Steering on all wheels also minimizes side slip of the wheels resulting in reduced wear on the vehicle and less damage to the field. Also, AgriBOT is a vehicle whose structure is oriented to automate his locomotion system and make it into an autonomous mobile robot using high level control systems to drive trajectory, therefore is necessary to do presentation about generalities related with fundamental AgriBOT topics to public he basic status, describing mechanics related to maneuverability, so there is propulsion and steering subsystems that depends on the operation of a electro-hydraulic

ISSN 2176-5480

deployment that is managed from a industrial programmable controller model MC050 commercialized by SAUER DANFOSS Company that is programmed from *Plus 1+ Guide* development tool.

Section 2 exposes about the general AgriBOT characteristic. Section 3 describes the basic concepts of CAN (Controller Area Network) protocol because the electronics subsystem use intensely this communication scheme and because for future deployments around control systems deploy will use CAN protocol. Section 4 describes the actual status of project. Section 5 presents conclusions and future work. Finally is presented the agreements and references.

2. AGRIBOT MÓBILE ROBOT

Figure 1. AgriBOT vehicle.

Figure 1 shows a picture of AgriBOT mobile robot, it is shown a 4WSD (Four Wheel Steering and Driven) vehicle. 4WSD configuration was chosen to gain a higher level of maneuverability due to the mobility capability provided by four identical independent modules that has electromechanical actuators to drive on steering and to acts on each wheel.

The vehicle chassis base is located 1,80m high from the ground. The distance between the wheels modules can be seen in figure 2, for lateral case this distances is adjustable from 2,25 m to 2,40 m. The research team has been decided to deploy a rectangular structure so that so that its longer side will locate in parallel with ways existing between traditional crops, thus the AgriBOT was development to moves positioning its wheels on the crop ways.

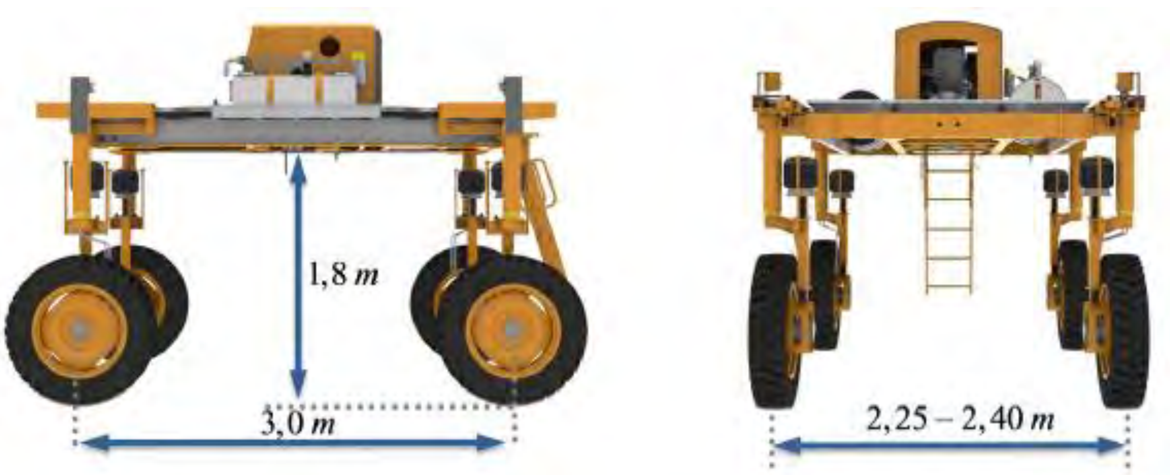

Figure 2. AgriBOT Vehicle Dimentions (Tabile, 2012).

Figure 3 shows the vehicle disposition of some components, these components are: 01 -- radiator; 02 -- Diesel motor; 03 -- Fuel tank; 04 -- secondary chassis; 05-- rotor coupling to the hydraulic circuit; 06 -- Wheel module; 07 – propulsion hydraulic motor; 08 -- railway gauge; 09 -- hydraulic pumps; 10 -- stairs; 11 -- hydraulic fluid reservoir; 12 -- Battery Pack.

ISSN 2176-5480

2.1 Steering and Propulsion Subsystems

AgriBOT's locomotion depends of four wheel identical modules. The wheel module shown in figure 4 allows to explain that to control the steering position of each wheel a 2-way hydraulic cylinder (03) assembled on top of the wheel shaft changes its position because the respective hydraulics is acting, this cylinder has attached a rack (10) that is modifying its linear position that transmit its displacement to a pinion (02) that transforms the linear displacement into rotary motion that is transferred to tire by the steering module shaft (01). According to the construction parameters, it is allowed to modify the steering angle for each wheel from the point in which it is aligned with the platform longitudinal axis, from 133º in the clockwise direction and 133º in the counterclockwise, totaling 266°.

Figure 3 shows basic structural components of a wheel module, as can be observed, this module is composed by a tractor tire (06) manufactured by Goodyear Inc.; a wheel rim (08); a hydraulic motor dedicated to vehicle propulsion (07); a pneumatic suspension subsystem (05); A rack (10); pinion (02); steering module shaft (01). There we have too a linear potentiometer (04) assembled to the rack to measures the rack displacement that is used to know the steering angle; Also, the wheel module has a pneumatic damper (05) and the mean chassis (09).

Wheel module accounts with a part of propulsion subsystem because it has the propulsion motor shown in figure 4. The basics aspects of propulsion subsystem were explained in paragraph 4 of section 2. So, Each wheel module provides two degrees of freedom, this structural and dynamic characteristic to allows the vehicle driving in special situations such as sharp curves and a good tolerance to hard changes of vehicle orientation, so, 4WSD configuration allows vehicle orientation changes that are impossible to do with vehicles that has rear or front wheels without some steering freedom (Sampaio et Al, 2011).

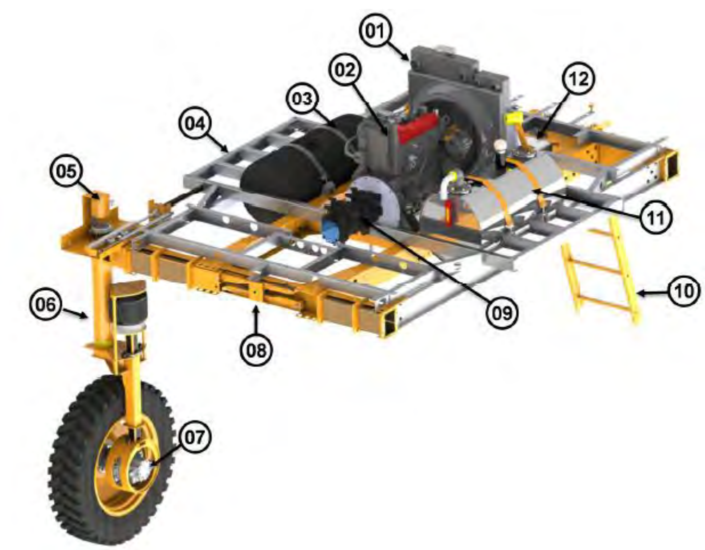

Figure 3. Vista geral dos principais componentes da plataforma (Tabile, 2012).

Figure 4. Wheel module (Tabile, 2012).

ISSN 2176-5480

2.2 Controller Area Network

Controller Area Network (CAN) is a serial network technology used to communicate Electronic Control Unit (ECU). CAN was originally development at the beginning of the 1980s by Bosch company and was designed for the automotive industry, but has also become a popular bus in industrial automation as well as other applications. CAN technology has been standardized since 1994 and is described by four ISO documents that indicate the details about hardware and software characteristics for the CAN deployment relating the layers of OSI model. (Voss, 2005; Hanxing et al, 2009).

A CAN network consists of a number of CAN nodes which are linked via a physical transmission medium (CAN bus). In practice, the CAN network is usually based on a line topology with a linear bus to which a number of electronic control units are each connected via a CAN interface. Figure 3 allows explaining easily the CAN basic concept. On picture is shown a vehicle in which a CAN network has 9 CAN nodes, like is depicted CAN nodes are connected to a CAN bus and each ECU has specific functions like monitor or control a device, in this example we assume that ECU 2 has the necessary electronics to measures the steering angle of front right wheel, thus the respective collected data are placed on the CAN bus, in the same way, we can assume that ECU 1 was deployed to drive an actuator installed to steer the front right wheel, thus the required information to drive this actuator is taken from the CAN bus; Now we are going to assume that ECU 3 has embedded a digital controller that take the information provide by ECU 2 and put on CAN bus the data necessary to ECU 2, therefore ECU 1, ECU 2 and ECU 3 completes a control system but distributed components.

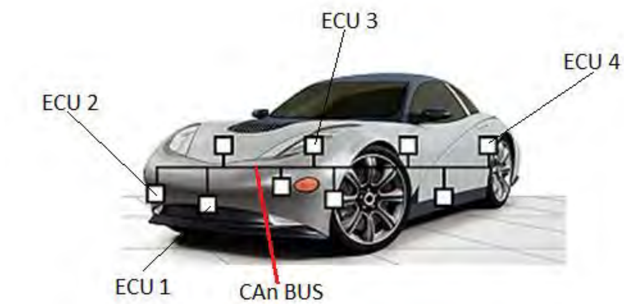

Figure 5. CAN network representation in a vehicle (Voss, 2005)

The CAN properties can be summarized as:

- Multi-Master priority based bus access
- Non-destructive contention-based arbitration
- Multicast message transfer by message acceptance filtering
- Remote data request
- Configuration flexibility
- System-wide data consistency
- Error detection and error signalling
- Automatic retransmission of messages that lost arbitration
- Automatic retransmission of messages that were destroyed by errors
- Distinction between temporary errors and permanent failures of nodes
- Autonomous deactivation of defective nodes.

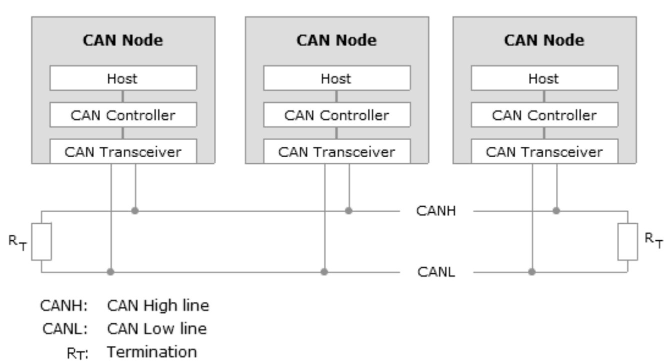

Figure 6. CAN nodes connected to CAN bus (Voss, 2005).

ISSN 2176-5480

A better representation of CAN network is show in figure 4 where is possible to observe that CAN bus is composed by two wires, a CAN High line (CAN-H) and CAN Low line (CAN-L), the bus needs a termination R_T. Also in figure 4 we CAN see that each CAN node has an organized architecture to participate in CAN communication, thus, a ECU has incorporated a *Host* a *CAN controller* and a *CAN transceiver*. The CAN controller full fills communication functions prescribed by the CAN protocol, which relieves the *host* considerably. The CAN transceiver connects the CAN controller to the physical transmission medium (Voss, 2005; Hanxing et al, 2009).

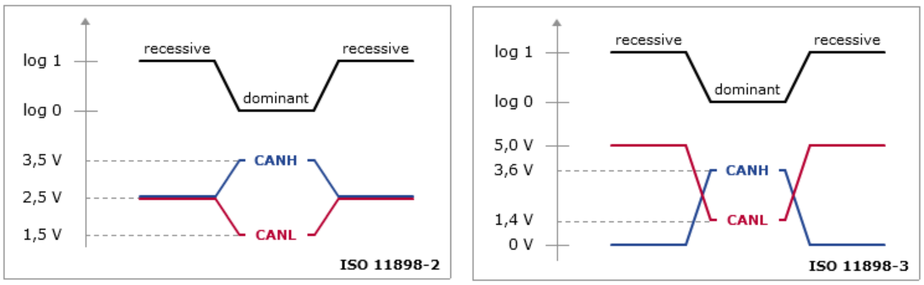

Figure 7. (a) ISO 11898-2 CAN bus voltage levels; (b) ISO 11898-3 CAN bus voltage levels. (Voss, 2005)

Figure 5 allows represents that physical signal transmission in a CAN network is based on differential signal transmission. The specific differential voltages depend on the bus interface that is used. A distinction is made here between the high-speed CAN bus interface (ISO 11898-2) and the low-speed bus interface (ISO 11898-3).

ISO 11898-2 assigns logical “1” to a differential voltage of 0 Volt. A differential voltage of 2 Volt signifies logical “0”. High-speed CAN transceivers interpret a differential voltage of more than 0.9 Volt as a dominant level within the common mode operating range, typically between 12 Volt and -12 Volt.

Below 0.5 Volt, however, the differential voltage is interpreted as a recessive level. A hysteresis circuit increases immunity to interference voltages. ISO 11898-3 assigns a differential voltage of 5 Volt to logical “1”, and a differential voltage of 2 Volt corresponds to logical “0”.

The dominant bus level corresponds to logical “0”. The recessive bus level corresponds to logical “1”.

2.2.1 Message frame formats

CAN network communication implies serial information interchange between ECUs, this interchange is achieved writing and reading data frames on CAN bus, in this way CAN protocol defines the frame characteristics. In CAN, there are four different types of frames, defined according to their content and function.

- Data frames, which contain data information from a source to (possibly) multiple receivers.
- Remote frames, which are used to request transmission of a corresponding (same identifier) Data frame.
- Error frames, which are transmitted whenever a node on the network detects an error.
- Overload frames, which are used for flow control to request an additional time delay before the transmission of a Data or Remote frame.

2.2.1.1 Data frame

Data frames are used to transmit information between a source node and one or more receivers. Data frames do not use explicit addressing to identify the message receivers. Instead, each receiver node states the messages that will be received based on their information content, which is encoded in the *Identifier* field of the frame. There are two different formats for CAN messages, according to the type of message identifier that is used by the protocol. Standard frames are frames defined with an 11-bit *Identifier* field. Extended frames have been made available from version 2.0 of the protocol as frames with a 29-bit *Identifier* field. Standard and extended frames can be transmitted on the same bus by different nodes or by the same node.

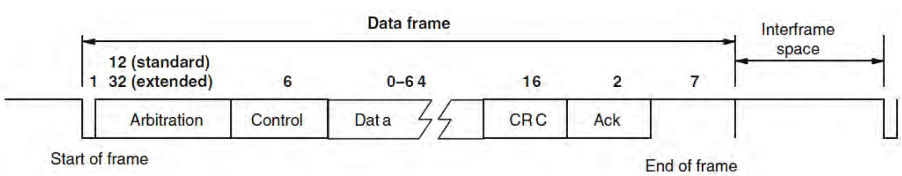

Figure 8. The CAN data frame format (Voss, 2005)

ISSN 2176-5480

The arbitration part of the protocol works regardless of the identifier version of the transmitted frames, allowing 29-bit identifier messages to be transmitted on the same network together with others with an 11-bit identifier. The CAN data frame format is shown in Figure 6, where the size of the fields is expressed in bits. Each frame starts with a single dominant bit, interrupting the recessive state of the idle bus. Then, the identifier field defines both the priority of the message for arbitration and the data content (identification) of the message stream. The other fields are: the *Control field*, containing information on the type of message; the *Data* field, containing the actual data to be transmitted, up to a maximum of 8 bytes; the *Checksum*, used to check the correctness of the message bits; the *Acknowledge* (ACK), used to acknowledge the reception; the *Ending delimiter* (ED), used to define the end of the message and the *Idle space* (IS) or *Interframe bits* (IF), used to separate a frame from the following one.

2.2.1.2 Identifier Field

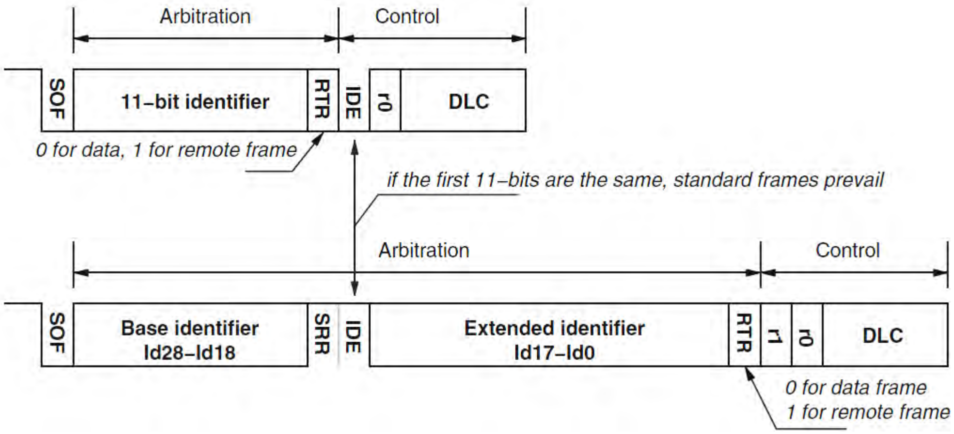

Figure. 9. The CAN identifier format

CAN protocol requires that all contending messages for the same medium have a unique identifier. The Identifier field consists of 11 (C1) bits in the standard format and 29 (C3) bits in extended format, following the scheme of Figure 7, in both cases, the Identifier field starts with the 11 bits (the most significant bits, in the extended format) of the identifier, followed by the Remote Transmission Request (RTR) bit in the standard format and by the Substitute Remote Request (SRR) in the extended format. The RTR bit distinguishes data frames from remote request frames. It is dominant for data frames, recessive for remote frames. The SRR is only a placeholder (always recessive) for guaranteeing the deterministic resolution of the arbitration between standard and extended frames.

2.2.1.3 Control Field

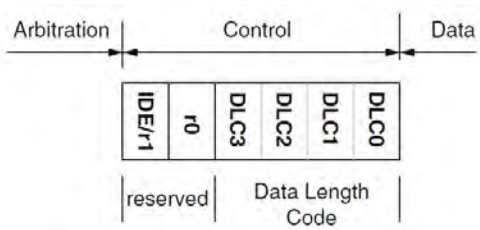

Figure. 10. The control field format. (Voss, 2005)

The *Control* field contains 6 bits. The first two bits are reserved or predefined in content. In the standard message format the first bit is the IDE (Identifier Extension bit), followed by a reserved bit. The IDE bit is dominant in standard formats and recessive in extended formats. It ensures the deterministic resolution of the contention (in favor of standard frames) when the first eleven identifier bits of two messages (one standard, one extended) are the same. In the extended format there are two reserved bits. For these bits, the standard specifies that they are to be sent as recessive, but receivers will accept any value (dominant or recessive). The following four bits in the data frame define the length of the data content (Data Length Content, or DLC) in bytes. If the dominant bit is interpreted as 1 (contrary to the common notation in which it is read as 0) and the recessive as 0, the four DLC bits are the unsigned binary coding of the length (Figure 8).

ISSN 2176-5480

2.2.1.4 CRC + ACK Fields

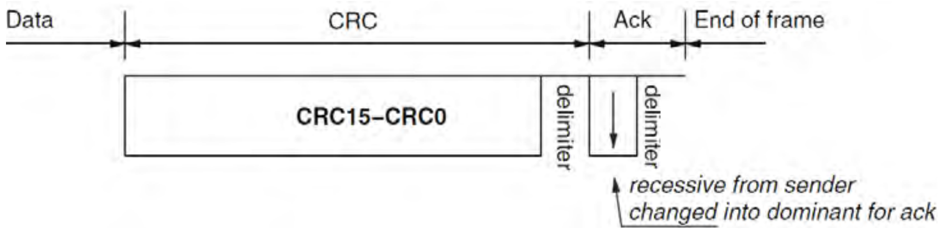

Figure. 11. The CRC and acknowledgement formats

The CRC and ACK fields are the next frame content. Their general layout is in Figure 9. The CRC portion of the frame is obtained by selecting the input polynomial (to be divided) as the stream of bits from the start of frame (SOF) bit (included) to the *Data* field (if present) followed by 15 zeros. This polynomial is divided by the generator

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

The remainder of the division is the CRC sequence portion of the frame. Details on the computation of the CRC field (including the generating algorithm) can be found in the Bosch specification. The ACK field consists of two bits. The first bit has the function of recording acknowledgments from receivers (ACK slot). The other bit is a delimiter (one bit of bus recessive state). The receivers that have validated the received message signal their acknowledgement to the sender by overwriting the recessive bit sent in the ACK slot by the transmitter with a dominant bit.

2.2.1.5 Interframe Space

Data frames and remote frames are separated by an Interframe space (7 recessive bits) on the bus. The frame segments Start of frame (SOF), Identifier, Control, Data and CRC are subject to bit stuffing. The remaining bit fields of the data frame or remote frame (CRC delimiter, ACK, and End of Frame (EOF) fields) are in fixed form and not stuffed.

2.2.2 Remote Frame

A remote frame is used to request the transmission of a message with a given identifier from a remote node. A remote frame has the same format of a data frame with the following characteristics:

- The Identifier field is used to indicate the identifier of the requested message.
- The Data field is always empty (0 bytes).
- The DLC field indicates the data length of the requested message (not the transmitted one).
- The RTR bit in the arbitration field is always set to be recessive.

2.2.3 Error Frame

The Error frame is not a real frame, but rather the result of error signaling and recovery action(s). The details of the error frame and its management are described in the following section on error management. The Error frame (not surprisingly given that it consists of 6 dominant or recessive bits in a sequence) is of fixed form with no bit stuffing.

2.2.4 Overload Frame

The purpose of the Overload frame is to inject a bus state at the end of a frame transmission that prevents the start of a new contention and then transmission. This allows an overloaded receiver to force a waiting period for the sender until it is once again ready to process the incoming data. The overload frame consists of 6 consecutive dominant bits. Its transmission starts during the first two bits of the Interframe space closing the transmission of the preceding frame. Since the interframe space should normally be of 3 bits, the overload signaling is detected by all other nodes, which also transmit overloads flags, possibly overlapping, up to a maximum of 12 dominant bits.

ISSN 2176-5480

Newer controllers typically don't need to send overload frames, but the standard requires that they are capable of understanding and reacting to an overload frame. Of course, the Overload frame is also of fixed form with no stuffing.

2.2.5 Bus Arbitration

The CAN arbitration protocol is both priority-based and non-preemptive, as a message that is being transmitted cannot be preempted by higher priority messages that are made available at the network adapters after the transmission has started. The CAN 2.0b Standard the CAN bus essentially works as wired-AND channel connecting all nodes. The media access protocol works by alternating contention and transmission phases. The contention and transmission phases take place during the digital transmission of the frame bits. At any time, if a node wishing to transmit finds the shared medium in an idle state, it waits for the end of the current bit (as defined by its internal oscillator), and then starts an arbitration phase by issuing a start-of-frame bit (a dominant bus state). At this point in time, each node with a message to be transmitted can start the competition to get access to the shared medium. All nodes will synchronize on the SOF bit edge and then, when the Identifier field starts, they will serially transmit the identifier (priority) bits of the message they want to send, one bit for each slot, starting from the most significant ones. Collisions among identifier bits are resolved by the logical AND semantics and each node read the bus state while it transmits.

If a node reads its identifier bits on the medium without any change, it realizes it is the winner of the contention and is granted access to transmit the rest of the message, while the other nodes switch to a listening mode. In fact, if one of the bits is changed when reading it back from the medium, this means there is a higher priority (dominant bit) contending the medium and thus the message is withdrawn by the node that has lost the arbitration. The node stops transmitting the message bits and switches to listening mode only.

Clearly, according to this contention resolution protocol, the node with the lowest identifier is always the winner of the arbitration round and is transmitted next (the transmission stage is actually not even a distinct stage, given that the message frame is transmitted sequentially with the fields following the Identifier). Given that in a CAN system, at any time, there can be no two messages with the same identifier (this property is also referred to as unique purity), the arbitration is deterministic and also priority-based, given that the message identifier gives in this case an indication of the message priority (the lowest identifier always wins).

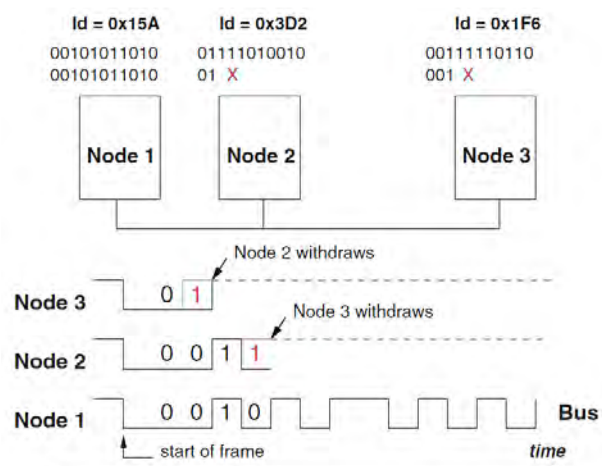

Figure 11. An example of CAN bus arbitration based on message identifiers (Voss, 2005)

An example of CAN bus arbitration is shown in Figure 10 three network nodes, labeled as Node 1 to Node 3 are trying to send the CAN messages with identifiers 0x15A, 0x3D2 and 0x1F6. The binary encoding is shown right below the hexadecimal message identifier values in the figure. The contention is started by the Start of Frame bit, followed by the identifier bits. As the first identifier bit, all nodes send a 0 (dominant value). As a result, the bus state also shows the 0 value, all nodes see no difference with the transmitted value and proceed. At the next bit, nodes 1 and 3 send a bit at 0, while node 2 sends a bit at 1. The bus state stays dominant at 0. Node 2 realizes it tried to send a 1 and read a 0; it stops the transmission of the identifier bits of its message and switches to listening mode. The next bit sent by Node 1 and Node 3 has value 1 for both. The bus state will be 1 and both nodes continue. On the next bit, node 1 sends a zero and node 3 sends 1. Given that the resulting bus state is 0, node 3 withdraws from the contention and node 1 is the winner; it keeps transmitting until the end of the identifier and follows with the remaining fields of the frame.

ISSN 2176-5480

3. AGRIBOT STATUS

AgriBOT is a vehicle whose principal power supply is a 4-stroke engine diesel turbo powered manufactured by Cummins Inc, that has a electronic fuel injection system that delivers 59.65 KW (80cv) at 2200 RPM. The fuel tank has a capacity of 140 liters of diesel. The Diesel Motor Power is transferred to the locomotion system using a hydrostatic transmission system.

AgriBOT maneuverability capabilities are reached by using the electro-hydraulic subsystem jointly with an electronic subsystem deployed. Figure 12 and 13 shows generics diagrams about the AgriBOTs electro-hydraulic deployment and general components related with mobility.

AgriBOT Propulsion subsystem depends on four wheel driving. This driving is achieved by operation of a hydraulic deployment by Bosch Rexroth AG Company. Propulsion Hydraulic subsystem has two variable displacement axial piston pumps (Pumps 1A and 1B) controlled by solenoid. Following figure 12 is possible to explain that each pump is responsible by act on two hydraulic motors and the vehicle stationary brake. Propulsion works in concept named by the research team as "*X configuration*" because as we can see again in figure 12, propulsion is reached controlling pump 1A to drives the motors 2 and 4 and controlling pump 1B is reached the drive of motors 1 and 3, thus, the vehicle has four hydraulic motors dedicated to propulsion, these motors are represented with the number 07 in figure 13. So, a hydraulic motor is set to each wheel axle, *enconder1* and *encoder 2* measures wheel angular velocity for wheels assembled to *motor 1* and *motor 4*.

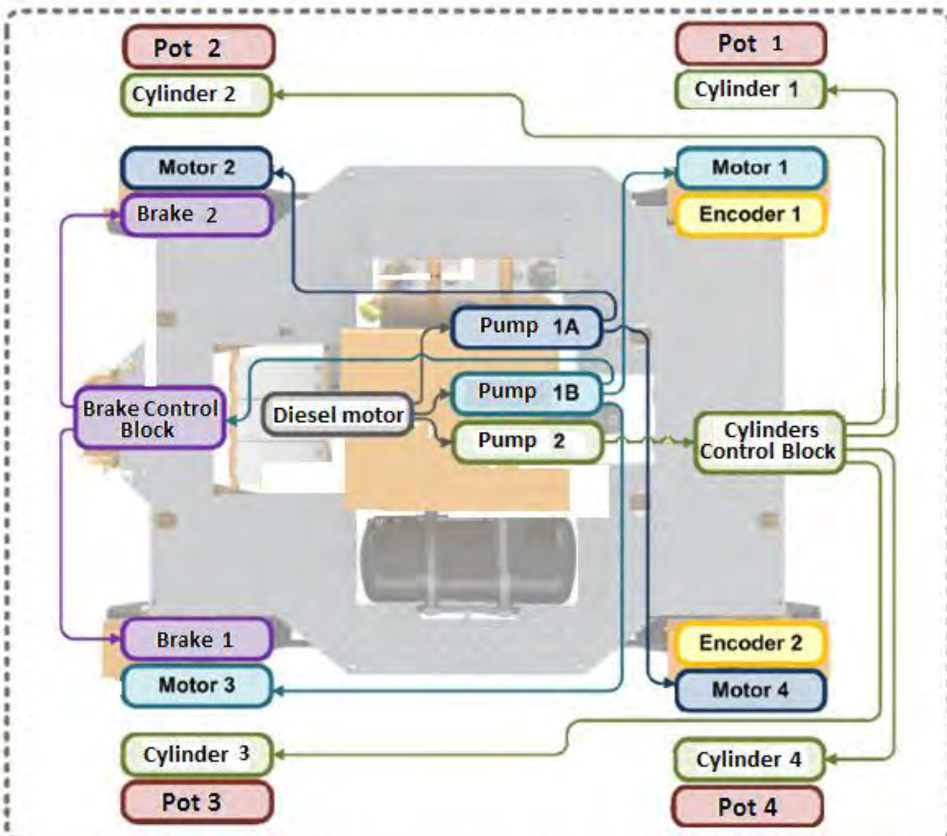

Figure 12. Generic representation of AgriBOT hydraulic system (Tabile, 2012).

Using one more time figure 12, is possible explain that wheels steering is achieved controlling pump 2 jointly a "*Cylinders Block Control*" composed by electro hydraulic valves installed manage each wheel steering independently; *Pot 1, Pot 2, Pot and Pot 4* are linear potentiometers that are used to measure each wheel steering angle.

AgriBOT has too a 12Vd.c, 510Ah electric subsystem power supply, consisting of three parallel connected batteries re-feeds for an alternator fixed to the diesel engine (Tabile, 2012; Tangerino et al, 2011).

Figure 13 allows explaining more about the subsystems related AgriBOT mobility and completes indications from figure 12. As wais exposed, Cumins Diesel motor generates power, part of these power is transferred to hydraulic subsystem and hydraulic subsystem drives propulsion and steering actuators. As can be seen in figure 13 AgriBOT has an industrial programmable controller model MC050 commercialized by SAUER DANFOSS Company that is

ISSN 2176-5480

programmed from *Plus 1+ Guide* development tool. MC050 has programmed routines to acts on diverse devices using for this purpose communication via a CAN bus, communication via J1939 bus and Pulse Width Modulated (PWM) outputs. The Cummins motor is handled directly from a Cummins Control Module built by motor factory and demands the use of J1939 communication to transmit the power on command and desired revolutions per second (RPM), therefore J1939 frames corresponding to power on and command desired RPM are programming in MC050. Propulsion subsystem was deployment by Bosch Rexroth AG Company and in consequence the research team hasn't access to modify this subsystem, in this way, from an external computer is communicated via CAN the desired propulsion commands that acts directly on pumps 1A and 1B. Encoders 1 and 2 shown in figure 12 measures angular velocities of wheels where are installed and its measurements are transmitted directly to *Bosch Module*. Information related to propulsion subsystem , *Bosch Module* puts on CAN bus. For Steering subsystem an external programmable device puts on CAN bus the respective frame to the desired steering angles of four wheels, MC050 receives this information and internally generates signals PWM to drive corresponding electro-valves that has its solenoids connected directly to PWM outputs of SAUER DANFOSS, potentiometers 1 to 4 uses voltage division principle to measures steering angles, the respective voltage because steering in on the same electrical node of MC0150 analogical inputs.

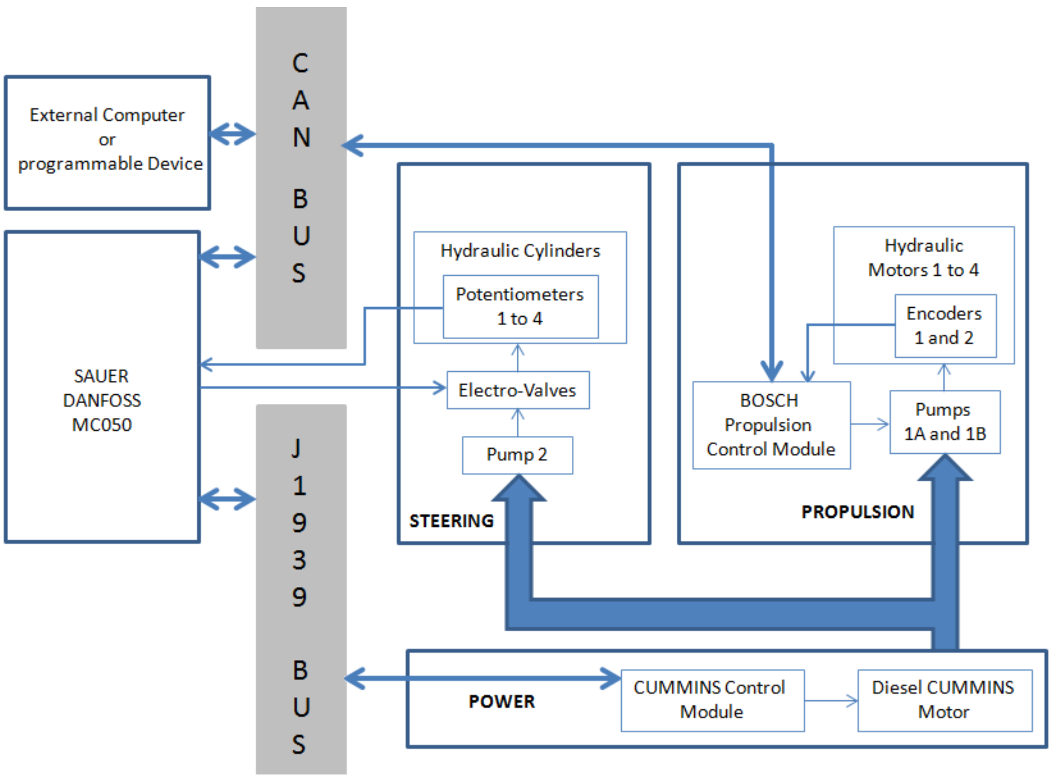

Figure 13. Block diagram related to AgriBOT maneuverability.

Actually research team accounts with a steering control system published in (Tangerino et AL, 2011), the controller was programmed on an commercial computer using the Development software tool famously commonly as LabView. Communication between computer and CAN bus was achieved using USB CAN 8473 interface.

Wheel module has a electro-valve between its components this valve allows to drive the steering , as is required counter clockwise and counterclockwise steering to each one of electro-valves correspond two solenoids to provides the respective hydraulics flux through its valves. Is possible to mention that steering control system is built in a closed loop like it is shown in figure 14. The mentioned closed loop control system integrates a commercial computer, an USB-CAN interface a MC050, electro-valve solenoids and potentiometers. δ_d represents the desired steering angle; ***e*** is a error signal result of comparison between the desired steering and the immediate real angular position of wheel, the steering angle is measured by linear potentiometer that as was mentioned is connected directly to MC050. MC050 has the appropriate electronics to transform potentiometers data from analog to digital, thus MC050 puts on CAN bus the potentiometers data. USB-CAN interface takes potentiometers information and provides it to comparator. Computer program made in LabView interprets error signal (***e***) and computes the control action (***u***) that is placed on CAN bus. Control action (***u***) is read from CAN bus by SAUER DANFOSS MC050 that translate the action control to PWM signals that are applied to solenoids and therefore drive valves and steering components of wheel module.

ISSN 2176-5480

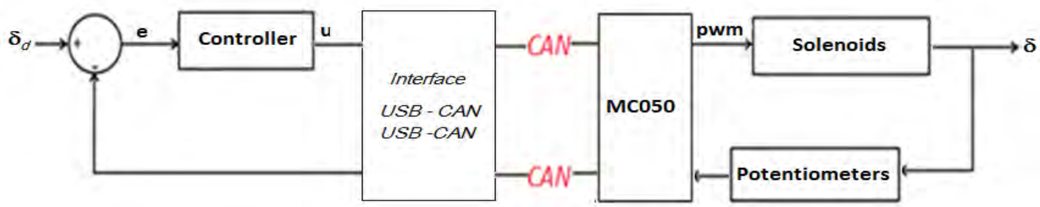

Figure 14. Closed loop control system representation for steering.

Diagram of figure 14 is a general representation to explain the closed loop control system used to control steering angles for four wheel modules of AgriBOT, details about control algorithm deployed can be encountered in (Tangerino et AL, 2011).

AgriBOT propulsion is achieved by a dedicated deployment development and installed by Bosch Rexroth AG Company, basically hydraulic motors transfers its rotational movement to respective wheel that because its contact with soil, so it is transformed the angular velocity (ω) to linear velocity and therefore the vehicle is propelled. Figure 15 show a blocks diagram to represent the control system about propulsion. The desired wheel angular velocity (ω_d) is informed to MOC050 from a external programmable device via CAN bus in direction to MC050 that transfer ω_d to CAN bus again but in direction to BOSCH propulsion control module that does the closed loop control work. Information about propulsion is putting on CAN bus by BOSCH propulsion control module in direction to SAUER DANFOSS that transfer this information to CAN bus again but in direction to external programmable device.

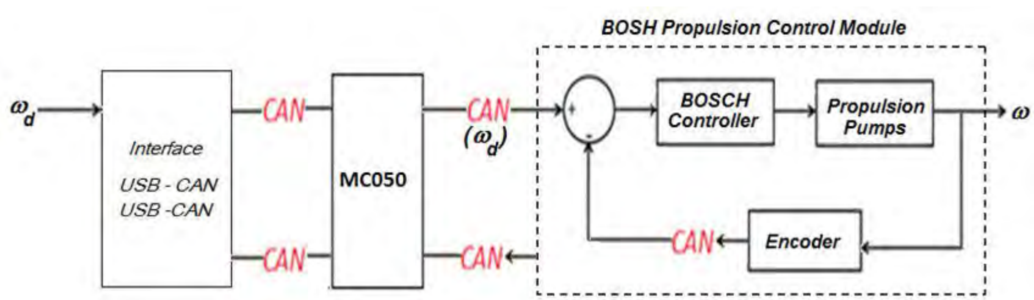

Figure 15. Block diagram representation for propulsion control system

4. CONCLUSIONS AND FUTURE WORK

AgriBOT is an agricultural vehicle which destiny is becomes an autonomous mobile robot to be employed on precision agriculture, for this purpose it has been proposed to assume in the near future that defiance from the perspective of a Robotic Control System where which has two levels as can be seen in figure 16. Low level control is the stage when steering and propulsion tasks are assumed. High level stage is proposed to deploy there all subsystems developed to autonomous navigation of AgriBOT, thus, the research team are researching on computational vision, auto localization using GPS (Global Positioning System) and Inertial Measurement Unit to provides information that will be processed to transfer the desired wheels steering angles (δ_d) and wheels desired angular velocities (ω_d) to low level control. So, research team is working to develop and deploy the high level control stage.

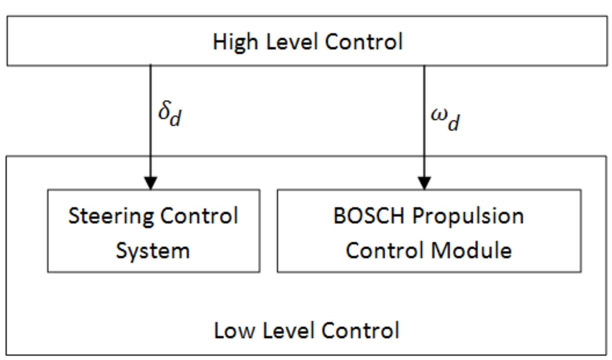

Figure 16. Proposed Robotic Control System.

For low level stage of figure 16, is possible to conclude that drive of wheels steering subsystem is available to be improved because the loop for control system must be closed by the designer, also must to deploy the control strategy, so, the designer has possibility of change control strategy if necessary. Wheels propulsion subsystem is closed, so, designer only can to report desired angular velocities.

5. ACKNOWLEDGEMENTS

Authors would like to thank FINEP, CNPq, CAPES, COLCIENCIAS (PhD Scholarship), EESCUSP, EMBRAPA support in this work.

6. REFERENCES

Sampaio, R.C.B., Borrero, H.G. and Becker, M., 2011. "Optimal H∞ controller on the adjustment of the steering angles of a two-track 4WD autonomous vehicle". In *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*. Bogotá, Colombia.

Tangerino, G., Godoy, E., Tabile, R., Souza, R. and Inamasu, R., 2011, "Controle de Esterçamento de Robô Agrícola Móvel de Quatro Rodas Guiáveis", X Simpósio Brasileiro de Automação Inteligente (SBAI).

Tabile, R., 2012. "*Desenvolvimento de uma plataforma robótica modular multifuncional para aquisição de dados em agricultura de precisão*". Tese de doutorado, Escola de Engenharia de São Carlos – Universidade de São Paulo, São Carlos – SP.

Voss, W., 2005. "A Comprehensible Guide to Controller Area Network", Copperhill Technologies Corporation, Greenfield, MA.

Hanxing Chen., Jun Tian, "Research on the Controller Area Network," Networking and Digital Society, 2009. ICNDS '09. International Conference on , vol.2, no., pp.251,254, 30-31 May 2009.

7. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.