IMPROVED COMMUNICATION BETWEEN MANUFACTURING ROBOTS

R. Stopforth[1*], **G. Bright**[2], **S. Davrajh**[3] and **A. Walker**[4]

[1,2,3,4]Mechatronics and Robotics Research Group (MR2G)
University of KwaZulu-Natal, South Africa
[1]stopforth@ukzn.ac.za, [2]bright@ukzn.ac.za, [3]sdavrajh@ukzn.ac.za, [4]awalker@ukzn.ac.za

ABSTRACT

Communication between manufacturing robots and autonomous vehicles in the industrial environment is important, sinceinstructions and information are crucial for communication between the control station and the robot station. Information is required between different manufacturing robots for optimal performance and dedication to industrial tasks within the environment. Failures in communication could cause robots to be a safety hazard or to perform tasks that are not required. This article shows how communication was improved with the use of the Robotics Communication Protocol (RCP) and an extension of this protocol.

OPSOMMING

Kommunikasie tussen vervaardigingsrobotte en outonome voertuie in 'n industriële omgewing is belangrik, aangesien opdragte en inligting krities is vir kommunikasie tussen die beheerstasie en die robotstasie. Inligting word benodig tussen verskillende vervaardigingsrobotte vir optimale werkverrigting en toewyding aan take in die omgewing. Mislukte kommunikasie mag veroorsaak dat robotte 'n veiligheidsrisiko word of veroorsaak dat onnodige take verrig word. Hierdie artikel toon hoe kommunikasie verbeter is deur die gebruik van die "robotika-kommunikasie-protokol" en 'n uitbreiding van die protokol.

*Corresponding author

1. INTRODUCTION

Communication is critical in the manufacturing and industrial environment, as the loss or incorrect transmission of data could cause a manufacturing robot to malfunction, perform incorrect tasks, or become a safety hazard in the environment. De Santis & Siciliano [1] conducted tests in industry robotic cells, finding that emergency stops were often activated due to communication problems. The resulting industry downtimes cause financial loss.

The poor reliability of communication between stations (whether the control station or the end-point station, such as a manufacturing robot) is caused mostly by interference and by structures in the environment. Machinery causes interference at lower frequencies, while the penetration of higher frequency signals used by wireless equipment is difficult through building materials.

Further interference is caused by equipment (like welding machines) that creates radiation. The fading of signals is also caused by reflection, refraction, and scattering. Signals are absorbed by water, humidity, and human bodies [2].

Hierarchy structures can allow for optimal performance, but problems could arise within the communication system with computational bottlenecks, reaction capability, and robustness [3].

Industrial robots used in the manufacturing environment must be wireless, as a tethered system could be a safety hazard or result in a limited controlled system. An example of an industrial robot that must be wireless is the autonomous self-balancing mobile materials handling platform [4]. These service vehicles allow for the production line to move between machinery and other robots.

The Robotics Communication Protocol (RCP) packet format is explained, with the operating procedures. An extension of this protocol is described, along with the use of two types of operating procedures that would typically be used in the manufacturing environment.

2. BACKGROUND

Interference with radio waves is mainly due to the robots' use of industrial, scientific and medical (ISM) bands which are unlicensed frequencies with certain constraints [5]. Many electronic communication units use the ISM bands. Interference can have life-threatening consequences, should a robot perform an unintended action. Dedicated frequencies will significantly prevent interference. The output power between the control unit and the robot can be constrained to prevent a signal from one unit overwhelming signals from other units.

Another reason for failed robot communication is the loss of signals between a robot and its control station. This is mainly caused by the frequency used. Signal penetration of buildings is also affected by the frequency used. Higher frequencies can penetrate denser materials that lower frequencies. The disadvantage of using higher frequencies is that small items, such as dust particles, resonate at a high frequency, absorbing the power of the signal. Therefore it is best to use a frequency in the centre, between the two extremes, allowing the best quality radio communication. UHF frequencies work best for this, as they can penetrate with a relatively low power output, and have a relatively good signal penetration.

3. PROTOCOLS

The use of protocols is important for data to be transmitted successfully. Using available protocols is an option, but performance and efficiency must be considered. Most existing protocols have been developed over many years and by various people, and have been optimised for specific tasks.

The IEEE 802.11 protocol could be used for communication between robots, but an access point is not always available for wireless communication. The communication between the robots will be ad hoc. Since UHF frequencies are used, the data rate will be lower compared with that used in wireless communication: Wireless communication systems, such as wireless internet, use frequencies in the 2.4 GHz band, and the quality of the bandwidth decreases as the frequency decreases. As the bandwidth decreases, additional collisions can occur; and so smaller packet sizes are needed. More data transmission from other stations is able to occur when the packet sizes are smaller.

3.1 Robot Communication Protocol (RCP)

The Robot Communication Protocol (RCP) [6] uses different fields and characteristics from the wired and wireless LAN protocols. The problem when using wireless communication technology is that it uses the 2.4 GHz band, causing the small particles in buildings to resonate at this frequency and to absorb energy – which in turn can prevent penetration through buildings. A further problem with the use of the IEEE 802.11 protocol is that its packets contain header details that are not used for some manufacturing robots, and take up bandwidth unnecessarily. Because the baud rate of the data communication modules can be low, unnecessary data must be eliminated, as this can saturate the medium.

Another problem with the existing protocols is that they may contain non-printable characters that cannot be processed by certain computers and microcontrollers. (Printable characters have an ASCII value between 31 and 127.)

A new wireless communication protocol is required for robots. A decision was made to use call signs to identify the robots and control units to prevent communication interference. A six-character call sign, using a combination of letters and numbers, is assigned to each robot and control unit. This makes $36^6 = 2.17 \times 10^9$ different call signs available.

There are two types of protocols that need to be transmitted: a 'one way packet' that is sent from one station to the other and that needs no confirmation (referred to from now on as a Robotic One-way (RO) packet); and a packet that is sent from one station to the other, and that replies with an acknowledgment of reception packet (a Robotic Confirmation (RC) packet).

There are four packets for the robotic network: Request-To-Send (RTS), Clear-To-Send (CTS), Acknowledgment (ACK), and Data. The different packets with their fields are explained below.

3.1.1 RTS / CTS / ACK packets

The packet formats for the RTS, CTS and ACK packets are shown in Figure 1.

Size	1 byte	1 byte	2 bytes	6 bytes	6 bytes	1 byte	1 byte
Field	Start	Type	Duration	RA	TA	Checksum	End

Figure 1: RTS / CTS / ACK packets

Start: The start character is for stations to identify the beginning of the packet. This is indicated with the hash (#) character. Should a station only start receiving in the middle of a transmission, it will recognise this and discard the packet. The reason for having a start byte is that the transmission is asynchronous on a single channel.

Type: This field indicates the type of packet that is being sent. The RTS, CTS, and ACK packets use the numbers 0, 1, and 2 respectively.

Duration: The duration of the transmission is specified in this field. This provides the other stations with the period to delay before attempting to transmit. The duration is specified by the number of characters. Time periods are calculated from the sum of the two bytes multiplied by x, where x is the time period for each character to transmit.

$$x = \frac{8\,bits}{baud\,rate} \tag{1}$$

Should these values be '#' or '!', then the most significant byte must be incremented and the least significant byte decremented.

RA: This is the address of the receiving station. This field allows other stations to identify whether or not that packet is for them. Should the packet not be intended for the station, the rest of the incoming packet can be disregarded and the station can start processing other incoming packets after the delay duration.

TA: This is the address of the transmitting station, used by the receiving station to identify whether the packet is from its approved station.

Checksum: This verifies the integrity of the packet. The field value consists of the sum of all ASCII values of all characters in packet modular 94 with the addition of 32. Should the receiving station receive a packet that is not approved, it is dropped. If the value of this field is equal to '#' or '!', the duration field is incremented and the checksum is recalculated. This field must be a printable character and not a control character (i.e. the character must have an ASCII value between 31 and 127).

End: This indicates the end of the packet with an exclamation mark (!).

3.1.2 Data packets

The format of the data packet is shown in Figure 2.

Size	1 byte	1 byte	2 bytes	6 bytes	6 bytes	0–255 bytes	1 byte	1 byte
Field	Start	Type	Duration	RA	TA	Data	Checksum	End

Figure 2: Data packet

Start: The start character is for stations to identify the beginning of the packet. This is indicated by the hash (#) character. In the event that a station only starts receiving in the middle of a transmission, this will be identified and the packet will be discarded. The reason for a start byte is that the transmission is asynchronous on a single channel.

Type: This field indicates the type of packet that is being sent. The identification of an RO data packet is the number 3, while for an RC data packet it is 4. The other possible values (except for the character values for # and !) for this field are reserved for future use.

Duration: The duration of the transmission is given here. This provides the other stations with the period that they have to delay before attempting to transmit. The duration is given by the number of characters. Time periods are calculated from the sum of the two bytes multiplied by x, where x is the time period for each character to transmit.

$$x = \frac{8\,bits}{baud\,rate} \tag{2}$$

Should these values be '#' or '!', then the most significant byte must be incremented and the least significant byte decremented.

RA: This is the address of the receiving station. This allows other stations to identify whether or not the packet is meant for them. If it is not, the station can ignore the rest of the incoming packet and start processing other incoming packets after the delay duration.

TA: This is the address of the transmitting station, used by the receiving station to identify that the packet is from its relevant approved station.

Data: The data for specific instruction or information between the stations is stored in this field. The only characters that are not allowed in this field are the hash (#) and the exclamation mark (!), as these are the start and end characters respectively. Control characters are also not allowed in this field.

Checksum: This verifies the integrity of the packet. The field value consists of the sum of all ASCII values of all characters in packet modular 94, with the addition of 32. Should the receiving station receive a packet that is not approved, it is dropped. If the value of this field is equal to '#' or '!', the duration field is incremented and the checksum is recalculated. This field must also be a printable character and not a control character (i.e. the character must have an ASCII value between 31 and 127).

End: An exclamation mark (!) indicates the end of the packet.

3.2 Communication procedure

The communication procedure is described using two stations: station A and station B. Should station A want to transmit, it observes whether any transmissions are occurring. If none are detected, then station A starts transmitting a RTS packet. All the stations in the vicinity of station A will delay transmission for the period of the duration field in the RTS packet. The delay duration period consists of the sum of the following:

- the time period needed to transmit the RTS packet
- the time period needed to transmit a CTS packet
- the time period for the data packet
- the time period to transmit an ACK packet (if this is needed)
- the sum of the processing time at each station

Station B receives the RTS packet and replies with a CTS packet containing a delay duration period, which is:

- the sum of the time period for the CTS packet
- the time period to transmit the data packet
- the time period to transmit an ACK packet (if this is needed)
- the sum of the processing time at each station.

Station A responds with the data packet containing a delay duration period, which is the sum of the time period for:

- the time period to transmit the data packet
- the time period to transmit an ACK packet (if this is needed)
- the sum of the processing time at each station.

Station B will reply with an ACK packet, should the last received packet have a type value of 100. This packet will contain a delay duration period, which is the sum of the time period to transmit the ACK packet and the processing time at each station.

Given that there is no stationary access point, there is no station that controls communication within the network. In Figure 3, four stations are shown with their respective radio coverage. C1 and R1 are control unit 1 and robot 1 respectively, and C2 and R2 are control unit 2 and robot 2 respectively.

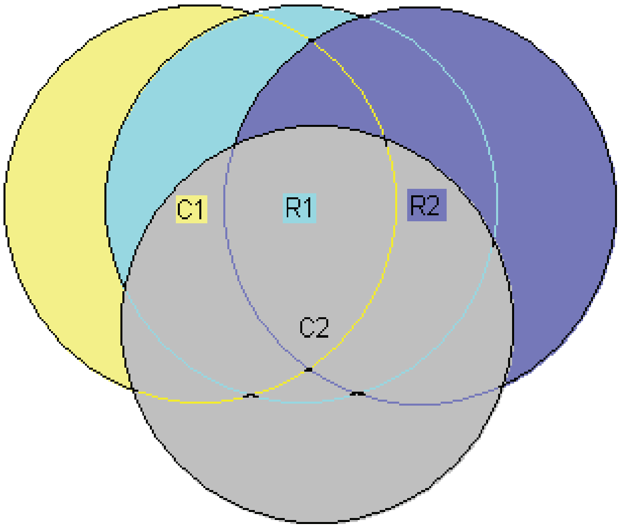

Figure 3: Radio coverage of two control units and two robots

As shown in Figure 3, C1 is in radio coverage with R1 and C2; R1 is in radio coverage with R2 and C2; R2 is in radio coverage with C2. Since C1 and R2 are not in radio coverage (transmission circle of C1 does not overlap the transmission circle of R2), packets to request transmission will not be received between these two stations. This is not a great disadvantage, as the different stations operate in an ad hoc system. The important point is that each robot is able to communicate with its own control unit.

Should an RTS packet be transmitted by C1, then R1 will receive the request and reply with a CTS packet. This CTS packet, which contains a duration field, will be received by R2 as well. Because R2 has received this packet, it will delay any transmission for this period before trying to transmit again.

Should C1 and C2 transmit an RTS at the same time, R1 will receive data that combines data from the two control units. R1 will reject this data, either because it will not recognise it or because it will not contain an acceptable packet. After a time-out period, C1 will realise that R1 has not responded and will transmit the RTS again if required.

As the RTS packets are relatively small, the retransmission overhead is small if two stations transmit the same time. The sum of data being sent in the data packet is limited to 128 characters, and it need not be sent in a specific format, provided that the format is understandable between the respective control units and robots.

The advantage of the RCP is that a computer system could be connected to a modem that uses the same protocol, and this modem could transmit and receive instructions and data to a large network of robots. In this situation the computer will be the control unit, and will not be dedicated to only a single robot. This network of robots could then be controlled to perform a task with a greater efficiency than a single robot could.

The RCP packets used to control a robot are at least 38% smaller than those used by hard-wired computer network protocols, and 33% smaller than those used by the IEEE 802.11 protocol. Communication between the robots and their control units is more reliable when used in a network. The use of a computer network protocol could be valuable when the robots have to transmit data and information that involves more than just the basic instructions.

3.3 Extension of the Robotic Communication Protocol

The current format and approach of the RCP needs modification to allow for manufacturing robots to communicate in an ad hoc network. The packet format remains the same, but with the addition of another field: time-to-live (TTL). This extension refers to the new version of the RCP [7]. The format of the extended packet is shown in Figure 4.

Size	1 byte	1 byte	1 byte	2 bytes	6 bytes	6 bytes	0–255 bytes	1 byte	1 byte
Field	Start	Type	TTL	Duration	RA	TA	Data	Checksum	End

Figure 4: Extended Robotics Communication Protocol format

TTL: This field contains the time-to-live value of the packet. The value of this field is decreased by a single value each time the packet is processed or relayed by a node, router, machine, or robot. Should the value of this field reach zero before the packet reaches the destination robot, the packet is discarded and not transmitted again.

The RTS, CTS, ACK, and data packets' format remains the same as that given in Figures 1 and 2. The only difference between the packets is that the data packet has a data field.

3.4 Communication procedure of the extended Robotic Communication Protocol

The extension to the RCP allows not only for manufacturing robots to communicate with each other, but also for the packets to be relayed between robots from the control station to the destination robot. There are two ways in which the manufacturing robots could communicate with each other. The different procedures are described with reference to Figure 3, considering that R2 needs to communicate with C1, and noting that no direct communication is possible between the two stations.

3.4.1 Direct transmission relay

R2 will transmit an RTS packet. R1 receives the packet, and as the packet is not meant for this station, it decrements the value in the TTL field and transmits the packet again. As soon as C1 has received the RTS packet and it is able to receive data, it transmits a CTS packet. R1 receives the packet, and as the packet is not meant for this station, it decrements the value in the TTL field and transmits the packet again.

After R2 has received the CTS packet, it returns a data packet. R1 receives the data packet, decrements the value of the TTL field, and transmits the packet again. C1 receives the packet and responds with an ACK packet. R1 receives the packet, decrements the value of the TTL field, and transmits the packet. R2 receives the ACK packet to confirm that the packet was successfully received. In the event that R2 does not receive the ACK packet in a determined time period, the process is restarted by sending C1 another RTS packet.

With the direct transmission relay procedure, all stations will delay the transmission of packets until the communication between R2 and C1 has been completed with an ACK packet. With this type of communication procedure, each station among the robots will be taking turns to transmit data.

3.4.2 Controlled line transmission relay

This transmission procedure is used between manufacturing robots that are positioned in a production line. Similar procedures to those in the original RCP are followed in relation to the relay station.

R2 transmits a RTS packet. R1 receives the RTS packet, and replies with a CTS packet. R2 responds with a data packet. After R1 has received the data packet, it stores it, and replies to R2 with an ACK packet.

R1 decrements the value in the TTL field within the data packet. R1 transmits a RTS packet. R2 ignores this packet as it is identical to a packet previously transmitted within a pre-determined time period. C1 receives the RTS and responds with a CTS packet. After R1 receives the CTS packet, it transmits the data packet. C1 receives the data packet and replies with an ACK packet. As the data packet is directed to C1, there is no need for C1 to transmit it again and therefore no need to decrement the value of the TTL field.

3.5 Modular approach for layer model

A layered model similar to the Open Systems Interconnection (OSI) model is needed for data communication. Each layer has a unique task in optimising the communication. The advantage of having a layered model is that each layer can be modified and optimised without affecting the other layers. The layered model can be represented in Figure 5.

Application
Data link / Transport / Session / Presentation
Physical

Figure 5: A three-layered model used by the Robotics Communication Protocol

This model has been divided into three layers, as each layer will be controlled by a separate module or microcontroller. The physical layer consists of the hardware that will be used – the radio modules that will act as the transceivers.

The layer that combines the data link, transport, session, and presentation is controlled by a single microcontroller. The data link layer is in control of the packets that are being sent, while the transport and session layer is responsible for the packet's control and transmission permission respectively. All the received data must be presented in a format that the computer understands. This is achieved by the presentation layer.

The application layer is involved in displaying the information, and in interacting with the user. This layer is also involved in the output, being the movement of the motors and any other of the robot's attachments. This layer will be controlled by a microcontroller which could be attached to other microcontrollers or modules, depending on the complexity of the attached module.

4. CONCLUSION

The original and extended versions of the Robotic Communication Protocol were tested with the use of a UHF modem. The UHF frequencies were found to penetrate manufacturing environments – something that was not possible with a wi-fi communication system. The headers of the packets allowed for dedicated communication procedures, preventing robots form executing instructions that were not meant for them.

Protocols have been explained, and the new robotic communication protocols, with their procedures of operation, have been given. The robotic protocol is 33% and 38% smaller than the IEEE 802.11 and hard-wired computer protocols respectively.

Extensions of the RCP for the relay of data between manufacturing robots are explained. The addition of the TTL field allows packets to be relayed in a direct line (as in a production line), or in an ad hoc network with a direct transmission relay.

The modular approach used with the RCP in conjunction with the implemented layered model has been explained.

The use of the RCP allows for reliable data communication within UHF frequencies between control stations and manufacturing robots; and the interaction allows for a swarm network, should this be required.

5. REFERENCES

[1] **De Santis, A. & Siciliano, B.** 2008. *Safety issues for human-robot co-operation in manufacturing systems, tools and perspectives in virtual machining*, Napoli.

[2] **Ladd, A., Bekris, K., Rudys, A., Wallach, D. & Kavraki, L.** 2004. On the feasibility of using wireless ethernet for indoor localization, *IEEE Transactions on Robotics and Automation*, Vol.20, No.3.

[3] **Henrich, D. & Honiger, T.** 1997. Parallel processing approaches in robotics, *IEEE International Symposium on Industrial Electronics*, Portugal.

[4] **Butler, L. & Bright, G.** 2008. Autonomous materials handling robot for reconfigurable manufacturing system, *24th International Conference on CAD/CAM, Automation, Robotics and Factories of the Future*, Japan.

[5] **Calcagno, R., Rusina, F., Sangiovanni, A. & Bonivento, A.** 2006. *Application of wireless technologies in automotive productive systems*, ISR-ROBOTIK 2006, Joint Conference on Robotics, Munich

[6] **Stopforth, R., Bright, G. & Harley, R.** 2008. Robotic communication protocol (RCP), *2nd Symposium in Mechatronics and Robotics*, Bloemfontein, South Africa.

[7] **Stopforth, R. & Bright, G.** 2010. An antenna for a mobile device and a communication system and methods. Patent No: 2009/ 03 009, South Africa.