ELEC 4309 Senior Design I

Robotic Communication

Nathan Schram

Due: 2017-DEC-20

Abstract

This paper will outline the different types of communication that exist between a robot and its controller (operator). It will look at the general difference between wired and wireless communication and go into more depth on the methods, interfaces, protocols of communication as well as the speeds and issues that must be overcome to achieve fluid control of the robot. We will then talk about the pitfalls that exist for wireless communication and products that exist on the market which can help overcome such obstacles. Finally, we will consider the differences that exist for analog and digital communication.

Analysis

Robots and robotic control have been around for decades. There applications are wide ranging from simple 1 axis manufacturing to complex multi-axis robotic arms that assist in moving cargo in a zero-gravity environment. There is no shortage of applications of robots and for this reason we see continued growth, advancement and development in this industry. As we progress with technology we also see smarter robots emerging that incorporate modern hardware and software that allow them to process information quicker and be more efficient. This trend is not going away as Moore’s Law indicates that every two years we are able to place twice as much silicon on a chip as the year before. This translates into twice as many transistors on the same area as was previously possible. As this trend continues we will continue to push and advance our technical prowess.

Robotic communication can first be classified into two main types of connection: wired and wireless. Wired connection is exactly what it sounds like. A physical, “tethered” connection between the robot and the controller. This means that any signals or commands being sent from the controller or operator must be done from very near the robot itself. Typically, this requires many cables running from the robot in a wire harness or wire bundle to the controller. The controller can also take many forms as the main source of all controls for the robot. But this is also determined by the complexity of the robot itself. If the robots main purpose is simple, a small custom-made controller might be made to keep the costs down. If the robot is sending information back to the controller for interpretation or data analytics, then the controller could be as sophisticated as a laptop. The controller can take many forms and shapes but typically is designed to fit the level of complexity required by the project. The second type of communication is wireless. This type of communication does not have any wires that directly connect the controller to the physical robot. Instead, a command structure is created and agreed upon on how to send and receive information between the two. Many different types of wireless communication are widely available and depending on the application many be appropriate. Two common types of wireless communication are WI-FI and Bluetooth.

Based on the complexity of the application the robot is being used for there are several advantages and disadvantages associated with wired vs. wireless robotic control. First, let’s look at wired communication. The advantages of wired communication are that you can process data much quicker across a hard line between the robot and the controller. You can also physically interact with the robot as you will likely be on-site and very near the robot, so if a human intervention is required, you will have someone very close to interact. Some of the disadvantages of wired connection are that you are limited to however long your wire harness is between the operator and the robot. The implied negative association is that you have physical lengths of wires that run to and from the controller. The more wires and connections you need to send back and forth, the more complex and ultimately the complexity of connection if there is any problem to debug while in the field. When you compare this with a wireless connection you can see that the distance restriction between the robot and the operator is much less restrictive. Using modern radio communication can move the point to point distance from operator to robot (assuming line of sight) to several miles. The implementation of this technique will be discussed later in the paper. Another advantage of wireless communication is that you can communicate to may robots in the same fashion. So, if there are more than one robot in the field you can communicate to each of them uniquely without having to have multiple sets of wire harnesses and controllers. This technique only needs one controller. However, this level of complexity often comes with a hidden cost which is the programming interface. For these more advanced control mechanisms a highly structured and rigorous communication algorithm must be designed and adhered to for fast data transmission. This is known as robotic communication protocol and can take many forms. What drives the protocol is typically what type of hardware is being used. We will look at what drives these protocols from a lower level next.

Robotic communication from controller to robot is typically a timed event, when things need to be collected at a set interval. For instance, if a temperature measurement needs to occur every 50ms. These determinations are usually set by the hardware that is on the robot. Typical hardware items are a microprocessor or microcontroller. A high-speed clock is typically connected on the PCB (printed circuit board) which drives the speed of which the communication is clock in at. Once these parameters have been determined, based on the requirements of the robot, processors architecture is also selected which has the required protocols needed for the application. There are many forms of low level communication protocols that exist on modern microprocessors. There are GPIO (general purpose input/out) pins, UART (universal asynchronous receiver/transmitter) pins, USART (universal synchronous and asynchronous receiver/transmitter), I2C (IIC) or SPI (serial peripheral interface) as well as many others. The number and type of low level communication is determined by the selection of the MCU. These interfaces must be analyzed prior to selecting the wireless communication standard for robotic control. The reason behind this is because there are set limits that data can be transferred on all low-level communication protocol. If you don’t take these speeds into account during your design phase you may end up trying to poll data that cannot be pushed at the same speed. When this occurs, you end up with other communicational problems such as data collision which is when data is being send from the controller to the robot and the robot is trying to send data to the controller at the same time. There are many ways to avoid this problem. One of the more commonly used low level communication protocols (SPI) can be used in two forms: half duplex and full duplex. These terms are used to discuss how data is transmitted. Half duplex mean data can only be sent one way at a time. If the transmitter is sending, then the receive may not do anything until the transmitter is complete. Full duplex means that data can be sent and received at the same time.

When SPI (full duplex) communication is used, data speeds can increase quite drastically to the point that an agreed upon protocol between the robot and the controller is required to make sure they are in sync and timed correctly. SPI can push data at approximately 400 MHz depending on what type of MCU is selected. This number is directly tied to the hardware and can be different from processor to processor. These speeds are still only being discussed at the lowest on-board hardware level. As we move higher up through the device we add layers of complexity. For one of the two common, wireless communication protocols mentioned (WI-FI or Bluetooth) there is additional low level hardware support that is required to use such technology. They both take the form of an integrated circuit (IC) which is put down on the PCB and physically connected to the host MCU via traces. Each IC has its own communication stack based on the technology being used. Each IC may have its own unique speed requirements for communicating with the MCU, which is yet another restriction for how fast data can be pushed/processed. The issue with these on-board (IC’s embedded on the PCB) is that they have very distinctive draw backs for long range communication. Each has its own set of requirements such as distance, current draw, max data rate, signal strength, etc. These parameters must be considered when designing a robot for a specific application.

In general, all forms of wireless communication have issues that are common among all wireless protocols. When data is sent over the air (OTA) the environment becomes critical to the success of the sent and received signal. If there are several obstacles, or you find that you are attempting to communicate across very long distances you will find that there are problems with this technique. Another layer of complexity for wireless data transmission is the time it takes for commands to be sent and received if you are in direct control of a robot. If there is a time sensitive time requirement for sending and receiving signals then we need to analyze what is called a communication link. A communication link is the time it takes for signals to get to their destination over a period with a minimum signal strength. This associated signal strength is important because if the signal arrives at the receiver in the correct time but the signal has degraded such that the signal is not identifiable by the robot then the signal is considered lost. When this occurs either a re-transmission is attempted or the lost signal is discarded and the robot waits for the next signal to arrive from the controller. RSSI (received signal strength indicator) Is typically associated with the reception of a signal from a source. This basically indicates if the signal received has sufficient strength for the robot to understand what to do from the controller. From a low level, this can be something as simple as an interrupt. If the signal’s RSSI was too low, an interrupt is generated telling the robot that there is an error. Another common term used in wireless transmission is latency. Latency is the time it takes for a signal to go from the controller to the robot. This can be a critical design aspect since timing often depends on strict agreed upon time frames. This design constraint may in in the form of a maximum allowable latency between the two devices. This would allow the receiver to know explicitly how long (maximum) it will take for a command to reach the robot. Obvious implications for this could be if for example a mars rover was heading for a cliff. The operator may need to send directional change signal to the robot to indicate a stop or change of direction to avoid having the robot go over the edge and destroy itself. If the maximum latency for communication between mission control and the robot is 15 mins. The operator better detect the cliff before he is within the latency period or the signal to adjust course will come in too late.

This example relies on the need to have human interaction with the robot. This is not always the case. There are many and more applications of autonomous robots (robots that do not need human interaction). These robots can have sophisticated algorithms to determine what it should do when faced with a decision. The more sophisticated a system the faster the execution time usually needs to be. Regardless of the application for the robot, speed of communication tends to be the lynchpin of the system. Questions that can be asked with respect to overall speed of communication can be how much of a data pipeline is required for real-time communication or control of a robot? What are the associated losses at such a speed? Does direct control (remote control) of the robot change the size and speed requirements of the data from the controller? In general, these questions can be answered based on the design requirements or design specifications. The type of data being sent is relevant as well as the distance the communications are being transmitted across. For a real-time control system to be in place, the data would need to be extremely fast as well as extremely reliable. These two parameters push the cost of the project up by requiring fast execution time of the system.

Let’s look a rover that is being remotely controlled by an operator approximately 5 miles away. For this type of distance there is only one method of control that is suitable for this application and that is RF (radio frequency). For this system, we would need to state that the latency for the transmission is small enough that control still is being done in real time. Wireless data can be sent over the air at a maximum rate of up to 4 Mbps (mega-bytes per second). Communication protocols are typically done by transferring bytes of data (8 bits per byte). Again, let’s assume that the time it takes for data to be delivered at 4 Mbps at a maximum distance of 5 miles is 10ms. Again, we need to analyze the link budget and describe something called the link margin. The link margin is the delta associated with the minimum signal strength required and the received signal strength. If the link margin is not satisfied that means the received signal is too weak to be of use to the system. There are several ways to improve signal strength from source to site such as higher gain antennas or improving the line of site of the two devices. However, we are not going to go into the physics of how to improve an RF signals gain in this paper. This example can be extrapolated by saying we want to send a video feed from the rover back to the controller. This problem changes the previous problem by the inclusion of a bandwidth for the data to be sent. In general, the more data you are trying to send requires a stronger signal and a fast communication protocol. If the receiver is trying to drive the robot based on the video feed this needs to be the highest priority in the robot’s design. If the distance of this information needs to change as well then, the link budget needs to be reanalyzed for a larger diameter area. Expanding the distance of the communication is not as difficult as it may seem.

Freewave makes several wireless products that are used for robotic control including P2P (point-to-point), P2MP (point-to-multi-point) and MP2MP (multi-point-to-multi-point) devices. In addition to those they also use repeaters. The communication protocol or choice for many of these devices is based on speed requirements. The two choices are serial communication and ethernet communication. They have serial radios that operate anywhere from 9400 baud to 115200 baud. For the ethernet radios, the speeds can easily go from 1Mbps to 4Mbps depending on the configuration and distance needed. As was previously mentioned, they use repeaters in a network to extend the range of the signal. However, the drawback for this is throughput. Every repeater you put in-between your source and your site cuts the throughput of the signal by 50%. The value that repeaters bring to a network is that you can cover vast distances by using one or two repeaters from a radio with a 60-mile distance. Therefore, assuming you have ample vertical mounting capabilities (towers) you can easily send and receive 200 miles away. Freeware’s radios are used for UAV’s and drones as well because they set up a mesh network where the vehicle moves into an adjacent node’s range as it moves. Again, this allows for continued communication to something that can be in flight for an extended period of time.

In this paper, I was briefly able to introduce the topic of robotic control both by looking at the pro’s and con’s associated with the difference between wired and wireless communication as well as looking at low level protocols that control how fast certain signal can be polled from an MCU. As you move from low level hardware higher through the device, you add complexity as different protocols can be used for short and long-range communication. We touched on Wi-Fi and Bluetooth for short range and RF communications (serial and ethernet) for long range. Programmed communication protocols are extremely detailed and system and requirement dependent documents that are based on the distance signals need to travel, the speed at which data needs to be sent and received and how many devices and what the calculated link budget is. In general, this is system and device dependent. Any of these introduced topics can become their own paper and was only briefly discussed on a topical level. Robotic communication is not my specialty. However, embedded system design and development is. I am currently working for Freewave as an Embedded Software Engineer working on programming long distance radios for robotic applications both for government and scientific research. The applications of which are long reaching. I find this work incredibility challenging and rewarding and am excited to put my education to work and explore the radio frequency applications of wireless communication protocol.